IMPROVED NUMERICAL SOLUTION OF MIXED H_2/H_∞ OPTIMIZATION WITH APPLICATIONS

THESIS

Linda D. Smith
Captain, USAF

AFIT/GAE/ENY/94D-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/94D-01

IMPROVED NUMERICAL SOLUTION OF MIXED H_2/H_∞ OPTIMIZATION WITH APPLICATIONS

THESIS
Linda D. Smith
Captain, USAF

AFIT/GAE/ENY/94D-01

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

AFIT/GAE/ENY/94D-01

IMPROVED NUMERICAL SOLUTION OF MIXED H_2/H_∞ OPTIMIZATION WITH APPLICATIONS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Linda D. Smith, B.S.
Captain, USAF

December 1994

Acknowledgements

I would first like to express my gratitude to all my fellow students in the aeronautical department. From the very beginning they have made this master's program fun and provided the support to keep me going. Also a special thanks to PhD graduate Dave Walker and PhD students Dave Jacques and Jim Brown. They answered many questions and provided insight into several problems addressed in this thesis.

Thank you also to my two advisors Dr. Brett Ridgely and Capt Bob Canfield. Their support and long hours gave me the encouragement to finish this thesis.

Finally, and most importantly, to my husband Scott. His patience and friendship helped me through every step of the Air Force Institute of Technology's program. From setting up programs on the computer to reading my thesis several times, he has always been there for me. Also to my parents who have always had the confidence in me to support me in any endeavor. I share this accomplishment with my husband and parents and know they made this possible.

Linda D. Smith

Table of Contents

Page

List of Figures

Figure Page

Figure Page

Figure Page

List of Tables

Table Page

AFIT/GAE/ENY/94D-01

Abstract

This study improved the numerical optimization technique used to compute mixed H_2/H_∞ controllers. The run time to generate the full curve of an H_2/H_∞ design was reduced by a factor of almost 200 for a specific example. This was accomplished by improving gradient calculations, upgrading the H_∞ norm calculation to capture multiple peaks in the singular value plot and using routines that minimized processing time. The optimization routine used was the MATLAB 'CONSTR' Sequential Quadratic Programming (SQP) routine. This SQP routine was compared to a FORTRAN SQP numerical optimizer, IMSL/DNCONG. The method was applied to an F-16 SISO example and a missile MIMO example. The mixed H_2/H_∞ controller for both these examples are compared to previous controller designs created using the LQG/LTR technique.

IMPROVED NUMERICAL SOLUTION OF MIXED H_2/H_∞ OPTIMIZATION WITH APPLICATIONS

I. Introduction

1.1 Objectives

This study will investigate improving computation of the mixed H_2/H_∞ controller using different optimization techniques. The following objectives will be addressed:

1. Reduce excessive run time and improve reliability of the optimization code.
2. Derive solutions for an F-16 SISO example.
3. Derive solutions for a missile MIMO example.
4. Compare MATLAB optimization routine to a FORTRAN IMSL/DNCONG routine.

1.2 Justification

Controller designs for aerospace systems, which are highly nonlinear, are often based on linear, time-invariant models. The primary purpose of a controller is to provide stability and performance in the presence of disturbances and uncertainties (i.e., perturbations or unmodeled dynamics). The challenge to the control designer is to develop a linear solution for these nonlinear systems. A multi-objective controller can be used to provide both stability and performance. One example of a multi-objective controller is the H_2/H_∞ controller. The H_2 part of the controller provides stability and performance in the face of noises acting on the system. The H_∞ part of the design is used to make the system behave as closely as possible to some ideal linear model (model matching) and provides for unmodeled dynamics and uncertainties in the model. The goal then is to find a controller which minimizes the effect of disturbances while providing an acceptable level of model matching and robustness.

The process of computing an optimal H_2/H_∞ controller is iterative. It is done by minimizing the two-norm of one system transfer function while constraining another's infinity-norm to be less than or equal to some given value. Many different values of the infinity-norm must be selected. The designer can then make tradeoffs in controller performance between noise rejection, tracking and uncertainties in the model. The desire is to provide a range of controllers between the optimal H_2 and optimal H_∞ controllers. With the capability to develop this range of controllers, the designer can decide what level of noise rejection versus what level of robustness to uncertainty is required for the system.

Over the past five years several optimization routines have been used to improve the process of generating a mixed controller. The first numerical optimization approach to find a mixed H_2/H_∞ controller was developed by Ridgely using a Davidon-Fletcher-Powell (DFP) routine [Rid91]. This routine used variable metrics, where information about previous iterations are stored and used to define a better search direction for the next iteration. This took several weeks to generate a set of mixed controllers when running on a SPARC 10 SUN workstation. There were also examples where the numerics caused such a problem that the mixed H_2/H_∞ controller could not be calculated for various infinity-norm constraints. Since then, other optimization routines such as penalty methods and conjugate gradient routines have been used on this controller design development. Currently, the MATLAB version of the Sequential Quadratic Programming (SQP) method is used. This method obtains a search direction by solving a quadratic objective function with linear constraints. The SQP method has cut the run time by a factor of 10 from the DFP routine.

There were cases where the infinity-norm gradients could not be accurately calculated for use in the SQP optimization routine. One of the causes of this problem was found to be multiple peaks in the singular value plot of the system. If the infinity-norm gradients can not be accurately calculated, the full range of controllers can not be designed. Without this full range of designs, the engineer is limited in the choice of controllers. Algorithms have been developed in this work that take into account the multiple peak problem and allow the calculation of the mixed controller.

There are two specific control design examples for which the mixed controllers could not be calculated at a full range of infinity-norm values. These include an F-16 Single-Input-Single-Output (SISO) normal acceleration command following controller and a controller for a tail/fin guided missile that is a Multiple-Input-Multiple-Output (MIMO) set-up. The algorithm developed to improve the accuracy of the H_∞ gradient will be used on these two examples.

The new algorithm is much faster and more reliable than the previous version. This provides the engineer with a more practical and complete method for designing controllers for uncertain (possibly nonlinear) systems operating in the face of disturbances.

1.3 Literature Review

The initial approach to the mixed objective problem was developed by Bernstein and Haddad [BH89]. Their approach was limited in that it minimized an upper bound to the two-norm rather than the two-norm itself. It also lacked the ability to define the order of the controller and to freely define the two-norm and infinity-norm transfer functions. Work done by Zhou [ZDB89], [ZDGB90] presents an approach where different exogenous inputs were allowed, but only one controller output. Yeh [YBC90] showed this to be the dual of [BH89].

The mixed controller design problem of [BH89] was transformed into a convex numerical optimization program by Khargonekar and Rotea [KR91]. This approach is an effective alternative to solving coupled nonlinear equations required by the method developed by [BH89]. Several software routines are currently available to solve convex optimization problems. Rotea and Khargonekar were the first to develop a true mixed H_2/H_∞ solution, valid only for a special case. They did not use an upper bound to the two-norm but rather the two-norm itself. They also allowed for two sets of exogenous inputs and controlled outputs. The one restriction of their development was full state availability. Ridgely developed a numerical solution to nine necessary conditions for an optimal mixed H_2/H_∞ controller. He solved an unconstrained optimization problem for the controller state space matrices using the DFP algorithm [Rid91]. His work also lifted the restriction of full state availability and developed the conditions that an output feedback H_2/H_∞ controller must satisfy. The most recent contribution to the solution of this problem was by Walker [Wal94], who significantly reduced the time required to generate the H_2/H_∞ controller. He reformed the mixed H_2/H_∞ optimization design process into a constrained optimization problem with an H_∞ constraint. He then applied the MATLAB SQP method of optimization to the constrained nonlinear problem.

Walker identified the problem of the multiple peaks in the singular value plot as the reason some problems do not converge to a mixed controller [Wal94]. The infinity-norm is defined by doing a frequency search of the magnitude of the singular value plot. The point where the maximum occurs is the infinity-norm. The difficulty in calculating the infinity-norm gradients is created when there is more than one frequency at which the maximum occurs. The optimization routine does not know at which frequency the norm should be differentiated. It will continually shift back and forth from one frequency to another, and thus prevent convergence to a mixed solution.

Several types of examples have been used to demonstrate the mixed controller design approach. Luke [Luk93] used an example of an F-16 normal acceleration command following model. His results showed that controllers designed using mixed H_2/H_∞ provided excellent performance and robustness at orders less than the order of the conventional LQG/LTR control design. He also addressed the problem of a MIMO tail-fin controlled missile. He used the MATLAB SQP routine, but the problem became numerically ill conditioned with no useful mixed controllers calculated.

1.4 Approach

Reduction of the run time for development of the mixed H_2/H_∞ controller was attempted first by trying other optimization routines. The original software developed by Walker was run using the MATLAB optimization toolbox [Gra93]. MATLAB has one built-in optimization routine, which is 'CONSTR'. There are several other routines written in FORTRAN that are numerically well conditioned for this type of problem and that can provide faster run times. MATLAB can be linked to FORTRAN to provide the capability of running FORTRAN optimization routines. One of the FORTRAN routines that will be tested is the DNCONG routine [IMS89]. DNCONG is a double precision routine that uses the SQP algorithm with user supplied gradients. This will allow a comparison between the line search and Hessian update methods used in MATLAB SQP with the the methods used in FORTRAN IMSL SQP routine.

Other things were also done to optimize the run time. First, most of the time used to calculate the mixed controller is spent doing the frequency search to find the infinity-norm. This search can be optimized by doing one general search to find the peaks of the problem, and then reducing the search down to a smaller range of frequencies about those peaks. This should not only improve run times but should also improve the accuracy of the infinity-norm and its gradients. Second, an active set strategy can be applied to only calculate the derivatives of the constraint when the program determines that the constraint is active. This is estimated to be where the second largest amount of computer time is spent. Third, the run time may also be improved by defining less storage space and having fewer function calls. Finally, normalizing the objective and constraints should also improve the efficiency of the code.

The most important improvement to the optimization code was the correction of the gradients for the objectives and constraints. Modifications were made to the gradients to take into account the subdiagonal terms of the controller state space matrices once it was put into modal form. Corrections also included a change to the two-norm gradient when the mixed controller was not at the H_2 optimal design. Modifications were also made to the stability constraint gradient so that it could be calculated analytically instead of using finite differencing.

The problem of multiple peaks when calculating the H_∞ constraint was also addressed. The solution was to incorporate a routine that defines where the peaks occur. This can be done by checking the slopes of the singular value plot. Peaks can be identified wherever the slope of the singular value plot is zero, and the slope changes from positive to negative about that point. An active set strategy can then be applied to constrain the problem at each point where a peak is identified. Using separate constraints for each peak will prevent the optimizer from jumping back and forth between peaks and causing the problem to not converge.

Other improvements were made to the code to reduce the run time, prevent non-stabilizing controller designs, and provide the capability to step along the mixed H_2/H_∞ curve automatically. A function in MATLAB called FMIN was used to find maximum points in the singular value plot once a peak had been defined. This was much more efficient than the several hundred point search over the entire frequency range. Also, to prevent non-stabilizing solutions a penalty function was added to the objective function when the stability constraint was violated. This proved to be an effective means to prevent non-stabilizing designs. An autostep function was also developed that allows the designer to set a desired percentage change in two-norm and infinity-norm and the function automatically steps along the curve. This reduced the run time and the required interaction by the designer with the code.

Finally, the inability to produce a mixed solution for some specific examples will be addressed. The algorithms developed for the multiple peak problem will be used along with other improvements in the code to provide a full range of solutions to these cases. The SISO F-16 acceleration command following problem and the MIMO tail/fin controlled missile problem have been identified as cases with multiple peaks. Application of the algorithm should provide improvements in the capability of calculating the mixed controllers and reduce the time required to find them.

II. Background

This chapter discusses the theory used to develop the mixed H_2/H_∞ controller along with background of numerical optimization. The discussion on the mixed H_2/H_∞ controller is broken into three parts. The first part begins with a discussion on how the H_2 controller is developed and then the background of the H_∞ controller follows. After the separate design methods for H_2 and H_∞ have been presented, an explanation of how these two methods are combined to form a mixed design is given. The numerical optimization section gives a brief overview of optimization, presents the history of the numerical optimization of the mixed H_2/H_∞ controller and provides a discussion of the SQP optimization routine. The numerical optimization section also contains the methods used to calculate derivatives of the H_2 and H_∞ norms.

2.1 Mixed H_2/H_∞ Controller Design

2.1.1 H_2 Optimization. The H_2 method is a well defined process to design internally stabilizing controllers for linear dynamic systems. The objective of an H_2 design is to synthesize a controller which minimizes the energy of a system output to a zero-mean white Gaussian noise input. The H_2 method is a generalization of the standard Linear Quadratic Gaussian (LQG) problem.

The H_2 optimization process can be explained using the diagram shown in Figure 2.1. The linear time-invariant model of the system dynamics is defined by P. The compensator developed through H_2 design is represented by K. Inputs to the system are zero-mean, Unit intensity White Gaussian Noise (UWGN) shown as w. The measured values coming out of the system are shown as y and are input into the compensator. The control signal generated by the compensator is labeled u and is fed back into the system. The controlled output of the system is represented as z. The goal of this method is to design a K that minimizes the energy of the output z, given a (UWGN) w.

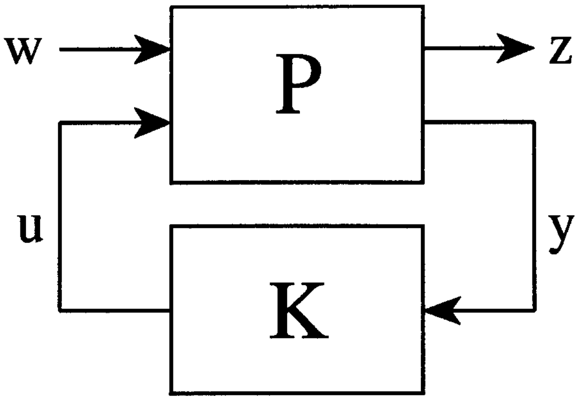

Figure 2.1 H_2 Feedback System

The plant represented as P in the diagram can be partitioned into its separate input and output transfer functions

$$P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} \tag{2.1}$$

where

$$\begin{aligned} z &= P_{zw}w + P_{zu}u \\ y &= P_{yw}w + P_{yu}u \end{aligned} \tag{2.2}$$

A state space realization of P can be written as

$$\begin{aligned} \dot{x_2} &= A_2x_2 + B_w w + B_{u_2}u \\ z &= C_z x_2 + D_{zw}w + D_{zu}u \\ y &= C_{y_2}x_2 + D_{yw}w + D_{yu}u \end{aligned} \tag{2.3}$$

Minimizing the two-norm of the closed-loop system is the same as minimizing the energy of the output of the system, given a (UWGN) input.

$$\underline{\alpha} = \inf_{K(s)Stabilizing} \|z\|_2 \quad \text{for } w \in UWGN \tag{2.4}$$

$$= \inf_{K(s)Stabilizing} \|T_{zw}\|_2 \tag{2.5}$$

$$= \inf_{K(s)Stabilizing} \|P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}\|_2 \tag{2.6}$$

The following assumptions are standard when designing an H_2 controller:

(i) $D_{zw} = 0$

(ii) $D_{yu} = 0$

(iii) (A_2, B_{u_2}) is stabilizable and (C_{y_2}, A_2) is detectable

(iv) $D_{zu}^T D_{zu}$full rank and $D_{yw} D_{yw}^T$full rank

(v) $\begin{bmatrix} A_2 - j\omega I & B_{u_2} \\ C_z & D_{zu} \end{bmatrix}$ has full column rank for all ω

(vi) $\begin{bmatrix} A_2 - j\omega I & B_w \\ C_{y_2} & D_{yw} \end{bmatrix}$ has full row rank for all ω

Assumption (i) is required so that the closed loop two-norm of the system is finite. Condition (ii) makes the development easier but can be removed completely. Requirement (iii) is required for existence of a stabilizing solution. This guarantees that the set of stabilizing controllers is not empty. Assumption (iv) is needed to prevent the use of inifinite control power and avoids the singular control problem. Finally, requirements (v) and (vi) are required to ensure the existence of solutions to the two Algebraic Ricatti Equations (ARE) required in the H_2 solution. These AREs are defined in (2.14) and (2.16).

All H_2 controllers can be represented by a Lower Fractional Transformation (LFT) of a transfer function J and a freedom parameter Q as shown in Figure 2.2. The freedom parameter is used to identify suboptimal H_2 controllers. These suboptimal controllers may be useful when used for computing designs that trade off H_2 performance for H_∞ performance. The mixed H_2/H_∞ optimization process will usually start at the H_2 optimal solution, where Q=0. Rescaling y and u so that condition (iv) is strengthened to

(iv)/ $D_{zu}^T D_{zu} = I$ and $D_{yw} D_{yw}^T = I$

results in the following H_2 optimal controller parameterization:

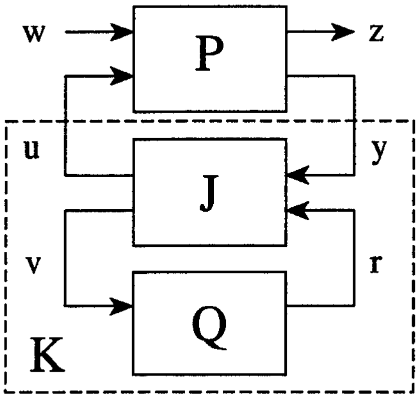

Figure 2.2 H_2 system with parameterized controller

$$J(s) := \begin{bmatrix} J_{uy} & J_{ur} \\ J_{vy} & J_{vr} \end{bmatrix} = \left[\begin{array}{c|cc} A_J & K_f & K_{fl} \\ \hline -K_c & 0 & I \\ K_{cl} & I & 0 \end{array}\right] \tag{2.7}$$

(2.8)

where

$$\begin{aligned} A_J &= A - K_f C_{y_2} - B_{u_2} K_c && (2.9) \\ K_c &= B_{u_2}^T X_2 + D_{zu}^T C_z && (2.10) \\ K_f &= Y_2 C_{y_2}^T + B_w D_{yw}^T && (2.11) \\ K_{cl} &= -C_{y_2} && (2.12) \\ K_{fl} &= B_{u_2} && (2.13) \end{aligned}$$

X_2 and Y_2 are the stabilizing solutions to the AREs

$$(A - B_{u_2} D_{zu}^T C_z)^T X_2 + X_2 (A - B_{u_2} D_{zu}^T C_z) - X_2 B_{u_2} B_{u_2}^T X_2 + \hat{C}_z^T \hat{C}_z = 0 \tag{2.14}$$

where

$$\hat{C}_z = (I - D_{zu} D_{zu}^T) C_z \tag{2.15}$$

and

$$(A - B_w D_{yw}^T C_{y_2})Y_2 + Y_2(A - B_w D_{yw}^T C_{y_2})^T - Y_2 C_{y_2}^T C_{y_2} Y_2 + \hat{B}_w \hat{B}_w^T = 0 \tag{2.16}$$

where

$$\hat{B}_w = B_w(I - D_{yw}^T D_{yw}) \tag{2.17}$$

If $Q = 0$, the controller is unique and optimal. If Q is nonzero, the set of all controllers such that $||T_{zw}|| \leq \alpha$ is parameterized by any $Q \in RH_2$ such that

$$||Q||_2^2 \leq \alpha^2 - \underline{\alpha}^2 \tag{2.18}$$

Another approach to minimize the two-norm of the system is through the use of the definition of the controllability grammian and a Lyapunov equation. Assume the state space closed-loop A matrix for the H_2 problem is stable. Define the controllability grammian as

$$L_c = \int_0^\infty e^{At} B B^T e^{At} dt \tag{2.19}$$

where L_c is the unique, real, symmetric positive semi-definite solution to the Lyapunov equation

$$0 = A L_c + L_c A^T + B B^T \tag{2.20}$$

The two-norm is then defined as

$$||G(j\omega)||_2^2 = tr[C L_c C^T] \tag{2.21}$$

This method was used for the mixed H_2/H_∞ optimization code. The Lyapunov equation is solved using the MATLAB command 'GRAM'. The objective function used for the optimization process is then the square of the two-norm, with the Lyapunov equation appended as a constraint.

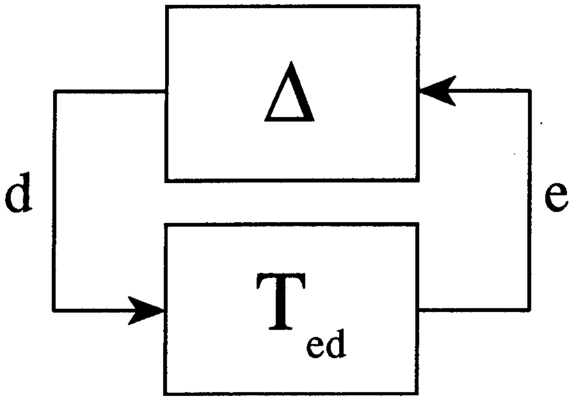

Figure 2.3   Perturbed stability diagram

2.1.2 H_∞ Optimization. The objective of an H_∞ controller is to minimize the energy of the output to the worst possible bounded energy input. This is the same as minimizing the infinity-norm of the transfer function. This type of design is robust to uncertainties in the plant. Let the uncertainty be called Δ. Figure 2.3 shows how the uncertainty would be introduced to the system. The question of how large a Δ can be introduced and the system remain stable is answered by use of the small gain theorem [Zam89]. For the closed loop system to be stable, Equation (2.22) must be satisfied.

Theorem 1 (Small Gain Theorem) *Let $T_{ed} \in H_\infty$. Assume $\Delta \in H_\infty$ is connected from e to d as shown in Figure 2.3. Then the closed-loop system is internally stable if*

$$||T_{ed}(s)\Delta(s)||_\infty \leq ||T_{ed}(s)||_\infty||\Delta(s)||_\infty < 1 \tag{2.22}$$

Proof: [Zam89]

Thus, by minimizing $||T_{ed}||_\infty$, the allowable $||\Delta||_\infty$ that cannot destabilize the system is maximized. H_∞ optimization is given by

$$\inf_{K_{adm}} \sup_{||d||_2 \leq 1} ||e||_2 = \inf_{K_{adm}} ||T_{ed}||_\infty \equiv \underline{\gamma} \tag{2.23}$$

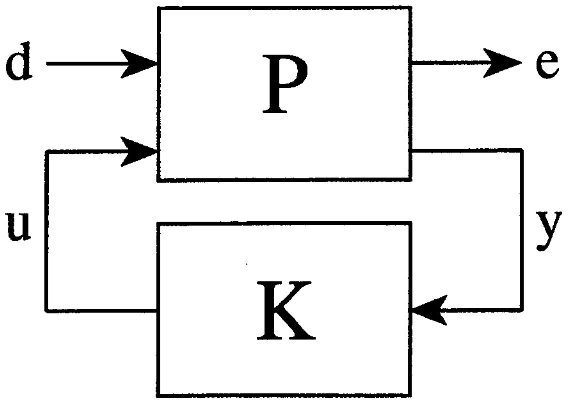

Figure 2.4 H_∞ Feedback System

where the infinity-norm of T_{ed} is

$$\|T_{ed}\|_\infty = \sup_\omega \bar{\sigma}[T_{ed}] \tag{2.24}$$

and $\bar{\sigma}$ denotes the maximum singular value. The minimum achievable infinity-norm, as indicated by Equation (2.23), is designated $\underline{\gamma}$.

The design diagram for H_∞ is the same as the figure for H_2 except for the definition of the inputs and outputs. Where the input to H_2 was a white noise labeled w, the input for H_∞ is a bounded energy exogenous input labeled d. The output for H_2 was a controlled output z, where the controlled output for H_∞ is labeled e. This diagram is shown in Figure 2.4.

The plant P in Figure 2.4 can be partitioned into its separate transfer functions as given by

$$P = \begin{bmatrix} P_{ed} & P_{eu} \\ P_{yd} & P_{yu} \end{bmatrix} \tag{2.25}$$

or

$$\begin{aligned} e &= P_{ed}d + P_{eu}u \\ y &= P_{yd}d + P_{yu}u \end{aligned} \tag{2.26}$$

A state space realization of the plant P is defined by

$$\begin{array}{rcccccc} \dot{x}_\infty & = & A_\infty x_\infty & + & B_d d & + & B_{u_\infty} u \\ e & = & C_e x_\infty & + & D_{ed} d & + & D_{eu} u \\ y & = & C_{y_\infty} x_\infty & + & D_{yd} d & + & D_{yu} u \end{array} \tag{2.27}$$

The following assumptions are made for the H_∞ design:

(i) $D_{ed} = 0$

(ii) $D_{yu} = 0$

(iii) (A_∞, B_{u_∞}) is stabilizable and (C_{y_∞}, A_∞) is detectable

(iv) $D_{eu}^T D_{eu} = I$ and $D_{yd} D_{yd}^T = I$ (strengthened through scaling)

(v) $\begin{bmatrix} A_\infty - j\omega I & B_{u_\infty} \\ C_e & D_{eu} \end{bmatrix}$ has full column rank for all ω

(vi) $\begin{bmatrix} A_\infty - j\omega I & B_d \\ C_{y_\infty} & D_{yd} \end{bmatrix}$ has full row rank for all ω

Conditions (i) and (ii) are not required for a solution to exist, but reduce the complexity of the solution. Condition (iii) is necessary for the existence of a stabilizing controller. Condition (iv) is a requirement that all control usage is penalized and no perfect measurements are allowed. Finally, conditions (v) and (vi) guarantee there is a solution to the Riccati equations for the H_∞ norm calculation.

The H_∞ controller is found by an iterative process. The H_∞ optimal controller found by this process is not unique. The family of all H_∞ compensators such that

$$\|T_{ed}\|_\infty < \gamma \tag{2.28}$$

is given by (see Figure 2.5)

$$K(s) = F_\ell\left[J(s), Q(s)\right] \tag{2.29}$$

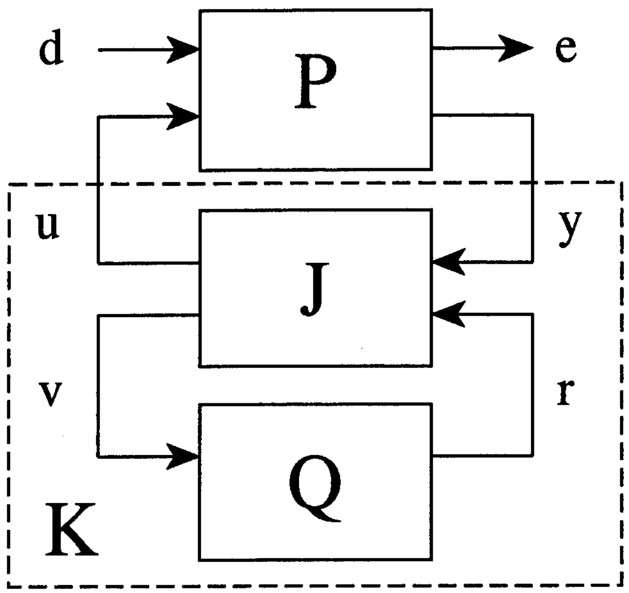

Figure 2.5 H_∞ system with parameterized controller

where

$$J(s) = \begin{bmatrix} J_{uy} & J_{ur} \\ J_{vy} & J_{vr} \end{bmatrix} = \left[\begin{array}{c|cc} A_J & K_f & K_{fl} \\ \hline -K_c & 0 & I \\ K_{cl} & I & 0 \end{array}\right] \tag{2.30}$$

and

$$A_J = A - K_f C_{y\infty} - B_{u\infty} K_c + \gamma^{-2} Y_\infty C_e^T (C_e - D_{eu} K_c) \tag{2.31}$$

$$K_c = (B_{u\infty} X_\infty + D_{eu}^T C_e)(I - \gamma^{-2} Y_\infty X_\infty)^{-1} \tag{2.32}$$

$$K_f = Y_\infty C_{y\infty}^T + B_d D_{yd}^T \tag{2.33}$$

$$K_{cl} = -(\gamma^{-2} D_{yd} B_d^T X_\infty + C_{y\infty})(I - \gamma^{-2} Y_\infty X_\infty)^{-1} \tag{2.34}$$

$$K_{fl} = \gamma^{-2} Y_\infty C_e^T D_{eu} + B_{u\infty} \tag{2.35}$$

The matrices X_∞ and Y_∞ are the solutions to the AREs

$$\begin{aligned} &(A - B_u D_{eu}^T C_e)^T X_\infty + X_\infty (A - B_{u\infty} D_{eu}^T C_e) \\ &\quad + X_\infty (\gamma^{-2} B_d B_d^T - B_{u\infty} B_{u\infty}^T) X_\infty + \hat{C}_e^T \hat{C}_e = 0 \end{aligned} \tag{2.36}$$

where

$$\hat{C}_e = (I - D_{eu} D_{eu}^T) C_e \tag{2.37}$$

and

$$\begin{aligned}(A - B_d D_{yd}^T C_{y\infty}) Y_\infty + Y_\infty (A - B_d D_{yd}^T C_{y\infty})^T \\ + Y_\infty (\gamma^{-2} C_e^T C_e - C_{y\infty}^T C_{y\infty}) Y_\infty + \hat{B}_d \hat{B}_d^T = 0\end{aligned} \tag{2.38}$$

where

$$\hat{B}_d = B_d (I - D_{yd}^T D_{yd}) \tag{2.39}$$

Finally, Q can be chosen to be any $Q \in H_\infty$ such that

$$\|Q\|_\infty < \gamma \tag{2.40}$$

The above parameterization of a controller K is valid if and only if the following three conditions hold:

(i) $H_X \in dom(Ric)$ with $X_\infty = Ric(H_X) \geq 0$

(ii) $H_Y \in dom(Ric)$ with $Y_\infty = Ric(H_Y) \geq 0$

(iii) $\rho(Y_\infty X_\infty) < \gamma^2$

The process of finding a controller which results in a closed-loop infinity-norm close to $\underline{\gamma}$ is iterative. To find a controller, select γ and check to see if it meets all three conditions above. If any condition fails, increase γ and repeat the check. If all three conditions are met, decrease γ. The desired $\underline{\gamma}$ can be found by this process to any desired accuracy.

2.1.3 Mixed H_2/H_∞ Optimization. A method was derived to achieve the robustness of H_∞ control with the noise rejection properties of H_2 by Ridgely [Rid91]. The mixed H_2/H_∞ diagram is shown in Figure 2.6. Using this method, weights can be added to both the

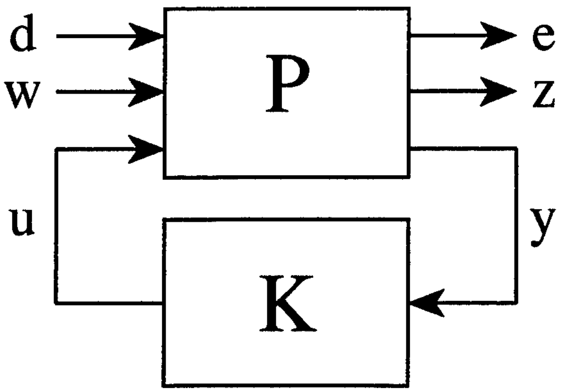

Figure 2.6 General mixed H_2/H_∞ optimization problem

H_2 and H_∞ set-up. The final goal of this mixed design process is to develop an admissible controller that satisfies the following:

$$\inf_{K stabilizing} \|T_{zw}\|_2 \text{ subject to } \|T_{ed}\|_\infty \leq \gamma$$

A general form of the mixed system can be written as

$$\begin{bmatrix} e \\ z \\ y \end{bmatrix} = \begin{bmatrix} P_{ed} & P_{ew} & P_{eu} \\ P_{zd} & P_{zw} & P_{zu} \\ P_{yd} & P_{yw} & P_{yu} \end{bmatrix} \begin{bmatrix} d \\ w \\ u \end{bmatrix} \tag{2.41}$$

The state space form of the system can be written as

$$\dot{x} = \tilde{A}x + \tilde{B}_d d + \tilde{B}_w w + \tilde{B}_u u \tag{2.42}$$

$$e = \tilde{C}_e x + \tilde{D}_{ed} d + \tilde{D}_{ew} w + \tilde{D}_{eu} u \tag{2.43}$$

$$z = \tilde{C}_z x + \tilde{D}_{zd} d + \tilde{D}_{zw} w + \tilde{D}_{zu} u \tag{2.44}$$

$$y = \tilde{C}_y x + \tilde{D}_{yd} d + \tilde{D}_{yw} w + \tilde{D}_{yu} u \tag{2.45}$$

The individual H_2 and H_∞ problems can be approached separately, as described in Subsections 2.1.1 and 2.1.2. The goal for H_2 is to find an admissible controller that minimizes the two-norm of T_{zw}, while the goal for H_∞ is to find an admissible controller for T_{ed} that has an infinity-norm less than or equal to some specified γ. The closed-loop transfer functions T_{ed} and T_{zw} for the mixed problem are defined as

$$T_{zw} = \mathcal{C}_z(sI - \mathcal{A}_2)^{-1}\mathcal{B}_w + \mathcal{D}_{zw} \tag{2.46}$$

$$T_{ed} = \mathcal{C}_e(sI - \mathcal{A}_\infty)^{-1}\mathcal{B}_d + \mathcal{D}_{ed} \tag{2.47}$$

The state space formulation of the mixed H_2/H_∞ controller is defined as:

$$\dot{x}_c = A_c x_c + B_c y \tag{2.48}$$

$$u = C_c x_c \tag{2.49}$$

The individual closed loop state space matrices can be represented by

$$\mathrm{x}_2 = \begin{bmatrix} x_2 \\ x_c \end{bmatrix} \tag{2.50}$$

$$\mathrm{x}_\infty = \begin{bmatrix} x_\infty \\ x_c \end{bmatrix} \tag{2.51}$$

$$\mathcal{A}_2 = \begin{bmatrix} A_2 & B_{u2}C_c \\ B_c C_{y2} & A_c \end{bmatrix} \tag{2.52}$$

$$\mathcal{A}_\infty = \begin{bmatrix} A_\infty & B_{u\infty}C_c \\ B_c C_{y\infty} & A_c \end{bmatrix} \tag{2.53}$$

$$\mathcal{B}_w = \begin{bmatrix} B_w \\ B_c D_{yw} \end{bmatrix} \tag{2.54}$$

$$\mathcal{B}_d = \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix} \quad (2.55)$$

$$\mathcal{C}_z = \begin{bmatrix} C_z & D_{zu} C_c \end{bmatrix} \quad (2.56)$$

$$\mathcal{C}_e = \begin{bmatrix} C_e & D_{eu} C_c \end{bmatrix} \quad (2.57)$$

$$\mathcal{D}_{zw} = 0 \quad (2.58)$$

$$\mathcal{D}_{ed} = D_{ed} \quad (2.59)$$

The assumptions for the mixed problem can simply be a combination of the assumptions for the separate H_2 and H_∞ problems. However, Walker [Wal94] has reduced the required assumptions to the following:

(i) $D_{zw} = 0$

(ii) $D_{yu} = 0$

(iii) (A_2, B_{u_2}) stabilizable, (C_{y_2}, A_2) detectable

(iv) $D_{zu}^T D_{zu}$ full rank, $D_{yw} D_{yw}^T$ full rank

(v) $\begin{bmatrix} A_2 - j\omega I & B_{u_2} \\ C_z & D_{zu} \end{bmatrix}$ has full column rank for all ω

(vi) $\begin{bmatrix} A_2 - j\omega I & B_w \\ C_{y_2} & D_{yw} \end{bmatrix}$ has full row rank for all ω

These are simply the assumptions for the H_2 problem. Since (i-vi) guarantee a strictly proper admissible controller, Walker found that the H_∞ assumptions could be relaxed. D_{ed} no longer must be zero and no assumptions need to be made on the ranks of D_{eu} and D_{yd} (i.e., singular and non-strictly proper H_∞ constraints are allowed).

The following definitions are made to simplify the development:

$$\underline{\gamma} := \inf_{K admissible} \|T_{ed}\|_\infty \quad (2.60)$$

$$
\begin{aligned}
\underline{\alpha} &:= \inf_{K admissible} \|T_{zw}\|_2 && (2.61)\\
K_{2_{opt}} &:= \text{the unique } K(s) \text{ that makes } \|T_{zw}\|_2 = \underline{\alpha} && (2.62)\\
\overline{\gamma} &:= \|T_{ed}\|_\infty \text{ when } K(s) = K_{2_{opt}} && (2.63)\\
K_{mix} &:= \text{a solution to the } H_2/H_\infty \text{ problem for some } \gamma > \underline{\gamma} && (2.64)\\
\gamma^* &:= \|T_{ed}\|_\infty \text{ when } K(s) = K_{mix} && (2.65)\\
\alpha^* &:= \|T_{zw}\|_2 \text{ when } K(s) = K_{mix} && (2.66)
\end{aligned}
$$

where "admissible" in (2.60) and (2.61) means stabilizing and at a fixed order. The conditions of the mixed H_2/H_∞ problem can now be stated as:

(i) $\mathcal{A}_2$ and $\mathcal{A}_\infty$ are stable

(ii) $\|T_{ed}\|_\infty \leq \gamma$ for some given $\gamma > \underline{\gamma}$

(iii) $\|T_{zw}\|_2$ is minimized.

The following theorem defines the problem further:

Theorem 2 *Let (A_c, B_c, C_c) be given and assume there exists a $Q_\infty = Q_\infty^T \geq 0$ satisfying*

$$\mathcal{A}_\infty Q_\infty + Q_\infty \mathcal{A}_\infty^T + (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T) R^{-1} (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T)^T + \mathcal{B}_d \mathcal{B}_d^T = 0 \quad (2.67)$$

where $R = (\gamma^2 I - \mathcal{D}_{ed}\mathcal{D}_{ed}^T) > 0$. Then the following are equivalent:

(i) $(\mathcal{A}_\infty, \mathcal{B}_d)$ is stabilizable

(ii) $\mathcal{A}_\infty$ is stable

(iii) $\mathcal{A}_2$ is stable.

Moreover, if the above hold then the following are true:

(iv) $\|T_{ed}\|_\infty \leq \gamma$

(v) the two-norm of the transfer function T_{zw} is given by

$$\|T_{zw}\|_2^2 = tr[\mathcal{C}_z Q_2 \mathcal{C}_z^T] = tr[Q_2 \mathcal{C}_z^T \mathcal{C}_z]$$

where $Q_2 = Q_2^T \geq 0$ is the solution to the Lyapunov equation

$$\mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2^T + \mathcal{B}_w \mathcal{B}_w^T = 0$$

(vi) all real symmetric solutions Q_∞ of (2.67) are positive semidefinite

(vii) there exists a unique minimal solution Q_∞ to (2.67) in the class of real symmetric solutions

(viii) Q_∞ is the minimal solution of (2.67) iff
$Re[\lambda_i(\mathcal{A}_\infty + \mathcal{B}_d \mathcal{D}_{ed}^T R^{-1} \mathcal{C}_e + Q_\infty \mathcal{C}_e^T R^{-1} \mathcal{C}_e)] \leq 0$ for all i

(ix) $\|T_{ed}\|_\infty < (\leq) \gamma$ iff $\Re e \left[\lambda_i(\mathcal{A}_\infty + \mathcal{B}_d \mathcal{D}_{ed}^T R^{-1} \mathcal{C}_e + Q_\infty \mathcal{C}_e^T R^{-1} \mathcal{C}_e)\right] < (\leq) 0$ where Q_∞ is the minimal solution to (2.67).

Proof: See [Wal94] Theorem (5.2.1)

The result of this theorem is that, given a controller which satisfies (2.67) [and thus bounds $\|T_{ed}\|_\infty$], we can determine the minimum α by also requiring tr[$C_z Q_2 C_z^T$] to be minimized. The mixed problem can then be rewritten as: determine the controller(A_c, B_c and C_c) that minimizes

$$J(A_c, B_c, C_c) = \|T_{zw}\|_2^2 = tr[Q_2 C_z^T C_z] \tag{2.68}$$

where Q_2 is the real symmetric, positive semidefinite solution to

$$\mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2^T + \mathcal{B}_w \mathcal{B}_w^T = 0 \tag{2.69}$$

and such that there exists a real, positive semi-definite solution to

$$\mathcal{A}_\infty Q_\infty + Q_\infty \mathcal{A}_\infty^T + (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T) R^{-1} (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T)^T + \mathcal{B}_d \mathcal{B}_d^T = 0 \tag{2.70}$$

The minimization problem now looks like an objective function, (2.68), and two constraint functions, (2.69) and (2.70). To solve this minimization problem a Lagrange multiplier approach is used. The Lagrangian is

$$\begin{aligned} \mathcal{L} &= tr[Q_2 \mathcal{C}_z^T \mathcal{C}_z] + tr\{[\mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2^T + \mathcal{B}_w \mathcal{B}_w^T] \mathcal{X}\} \\ &+ tr\{[\mathcal{A}_\infty Q_\infty + Q_\infty \mathcal{A}_\infty^T + (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T) R^{-1} (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T)^T \\ &+ \mathcal{B}_d \mathcal{B}_d^T] \mathcal{Y}\} \end{aligned} \tag{2.71}$$

This approach was introduced by Ridgely [Rid91] and modified by Walker [Wal94] to include the D_{ed} term and allow for the singular H_∞ constraint. The first order necessary conditions for the minimum of the Lagrangian (where $\mathcal{X}$ and $\mathcal{Y}$ are symmetric Lagrange multipliers) are

$$\frac{\partial \mathcal{L}}{\partial A_c} = X_{12}^T Q_{12} + X_2 Q_2 + Y_{12}^T Q_{ab} + Y_2 Q_b = 0 \tag{2.72}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathcal{B}_c} &= X_{12}^T Q_1 C_{y_2}^T + X_2 Q_{12}^T C_{y_2}^T + X_{12}^T V_{12} + X_2 B_c V_2 + Y_{12}^T Q_a C_{y\infty}^T \\ &+ Y_2 Q_{ab}^T C_{y\infty}^T + Y_{12}^T V_{ab} + Y_2 B_c V_b + (Y_{12}^T Q_a + Y_2 Q_{ab}^T) C_e^T M \\ &+ (Y_{12}^T Q_{ab} + Y_2 Q_b) C_c^T D_{eu}^T M = 0 \end{aligned} \tag{2.73}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial C_c} &= B_{u_2}^T X_1 Q_{12} + B_{u_2}^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2 + B_{u_\infty}^T Y_1 Q_{ab} \\ &+ B_{u_\infty}^T Y_{12} Q_b + R_{ab}^T Q_a Y_1 Q_{ab} + R_{ab}^T Q_a Y_{12} Q_b + R_{ab}^T Q_{ab} Y_{12}^T Q_{ab} \\ &+ R_{ab}^T Q_{ab} Y_2 Q_b + R_b C_c Q_{ab}^T Y_1 Q_{ab} + R_b C_c Q_b Y_{12}^T Q_{ab} \\ &+ R_b C_c Q_{ab}^T Y_{12} Q_b + R_b C_c Q_b Y_2 Q_b \\ &+ P_1 (Y_1 Q_{ab} + Y_{12} Q_b) + P_2 (Y_{12}^T Q_{ab} + Y_2 Q_b) = 0 \end{aligned} \tag{2.74}$$

$$\frac{\partial \mathcal{L}}{\partial \mathcal{X}} = \mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2^T + \mathcal{B}_w \mathcal{B}_w^T = 0 \tag{2.75}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial Q_2} &= \mathcal{A}_2^T \mathcal{X} + \mathcal{X}\mathcal{A}_2 + \mathcal{C}_z^T \mathcal{C}_z = 0 && (2.76) \\
\frac{\partial \mathcal{L}}{\partial \mathcal{Y}} &= \mathcal{A}_\infty Q_\infty + Q_\infty \mathcal{A}_\infty^T + (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T) R^{-1} (Q_\infty \mathcal{C}_e^T + \mathcal{B}_d \mathcal{D}_{ed}^T)^T \\
&\quad + \mathcal{B}_d \mathcal{B}_d^T = 0 && (2.77) \\
\frac{\partial \mathcal{L}}{\partial Q_\infty} &= (\mathcal{A}_\infty + \mathcal{B}_d \mathcal{D}_{ed}^T R^{-1} \mathcal{C}_e + Q_\infty \mathcal{C}_e^T R^{-1} \mathcal{C}_e)^T \mathcal{Y} \\
&\quad + \mathcal{Y}(\mathcal{A}_\infty + \mathcal{B}_d \mathcal{D}_{ed}^T R^{-1} \mathcal{C}_e + Q_\infty \mathcal{C}_e^T R^{-1} \mathcal{C}_e) = 0 && (2.78)
\end{aligned}$$

where

$$\begin{aligned}
M &= R^{-1} D_{ed} D_{yd}^T && (2.79) \\
P_1 &= D_{eu}^T R^{-1} D_{ed} B_d^T && (2.80) \\
P_2 &= D_{eu}^T M B_c^T && (2.81) \\
Q_2 &= \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} && (2.82) \\
\mathcal{X} &= \begin{bmatrix} X_1 & X_{12} \\ X_{12}^T & X_2 \end{bmatrix} && (2.83) \\
Q_\infty &= \begin{bmatrix} Q_a & Q_{ab} \\ Q_{ab}^T & Q_b \end{bmatrix} && (2.84) \\
\mathcal{Y} &= \begin{bmatrix} Y_1 & Y_{12} \\ Y_{12}^T & Y_2 \end{bmatrix} && (2.85) \\
\mathcal{B}_w \mathcal{B}_w^T &= \begin{bmatrix} B_w \\ B_c D_{yw} \end{bmatrix} \begin{bmatrix} B_w^T & D_{yw}^T B_c^T \end{bmatrix} \\
&= \begin{bmatrix} V_1 & V_{12} B_c^T \\ B_c V_{12}^T & B_c V_2 B_c^T \end{bmatrix} && (2.86) \\
\mathcal{B}_d (\mathcal{D}_{ed}^T R^{-1} \mathcal{D}_{ed} + I) \mathcal{B}_d^T &= \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix} (\mathcal{D}_{ed}^T R^{-1} \mathcal{D}_{ed} + I) \begin{bmatrix} B_d^T & D_{yd}^T B_c^T \end{bmatrix}
\end{aligned}$$

$$= \begin{bmatrix} V_a & V_{ab}B_c^T \\ B_cV_{ab}^T & B_cV_bB_c^T \end{bmatrix} \tag{2.87}$$

$$\mathcal{C}_z^T\mathcal{C}_z = \begin{bmatrix} C_z^T \\ C_c^TD_{zu}^T \end{bmatrix} \begin{bmatrix} C_z & D_{zu}C_c \end{bmatrix}$$

$$= \begin{bmatrix} R_1 & R_{12}C_c \\ C_c^TR_{12}^T & C_c^TR_2C_c \end{bmatrix} \tag{2.88}$$

$$\mathcal{C}_e^TR^{-1}\mathcal{C}_e = \begin{bmatrix} C_e^T \\ C_c^TD_{eu}^T \end{bmatrix} R^{-1} \begin{bmatrix} C_e & D_{eu}C_c \end{bmatrix}$$

$$= \begin{bmatrix} R_a & R_{ab}C_c \\ C_c^TR_{ab}^T & C_c^TR_bC_c \end{bmatrix} \tag{2.89}$$

Equations (2.75) and (2.77) are the original constraint functions. One solution to (2.78) is that $\mathcal{Y}$ equals zero; the other possibility is that $(\mathcal{A}_\infty + \mathcal{B}_d\mathcal{D}_{ed}^TR^{-1}\mathcal{C}_e + Q_\infty\mathcal{C}_e^TR^{-1}\mathcal{C}_e)$ is neutrally stable. The first solution corresponds to γ being off the boundary of the H_∞ constraint. The second solution corresponds to γ^* lying on the boundary of the H_∞ constraint. The off boundary condition of $\mathcal{Y} = 0$ results in the Lagrangian reducing to

$$\mathcal{L} = tr[Q_2\mathcal{C}_z^T\mathcal{C}_z] + tr\{[\mathcal{A}_2Q_2 + Q_2\mathcal{A}_2^T + \mathcal{B}_w\mathcal{B}_w^T]\mathcal{X}\} \tag{2.90}$$

which is just the H_2 Lagrangian; thus, for $\mathcal{Y} = 0$, $K_{mix} = K_{2opt}$. The following theorem was developed by Walker [Wal94].

Theorem 3 *Assume $n_c \geq n_2$, where n_c is the order of the controller and n_2 is the order of the plant for the H_2 problem. Then the following hold:*

1. *If $\gamma < \underline{\gamma}$, no solution to the mixed H_2/H_∞ problem exists*
2. *If $\underline{\gamma} < \gamma \leq \overline{\gamma}$, K_{mix} is such that $\gamma^* = \gamma$*
3. *If $\gamma \geq \overline{\gamma}$, $K_{2_{opt}}$ is the solution to the mixed H_2/H_∞ problem.*

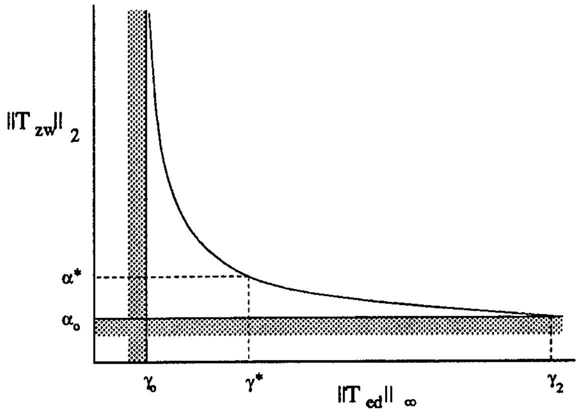

Figure 2.7 Typical H_2/H_∞ curve

In summary, the solution to the mixed H_2/H_∞ problem will lie on the H_∞ constraint boundary whenever the order of the controller is chosen greater than or equal to that of the H_2 problem. Theorem 3 implies that, for $\underline{\gamma} \leq \gamma \leq \overline{\gamma}$, each point on the mixed H_2/H_∞ curve occurs on the boundary of the H_∞ constraint. Therefore, between $\underline{\gamma}$ and $\overline{\gamma}$ the value of α^* is a smooth monotonically decreasing function as shown in Figure 2.7. This type of curve assumes the optimization process has found the global minimum of the mixed H_2/H_∞ problem or that the local minimum does not change between γ steps. Currently, the mixed problem can not be solved analytically, and must be solved through a numerical approach. The numerical approach is addressed in Section 2.2.

2.2 Numerical Optimization

2.2.1 Optimization Background. This section is a general overview of optimization theory. For details the reader is referred to [Van84]. The general nonlinear constrained optimization problem can be stated mathematically as:

$$\text{Minimize:} \quad F(X) \qquad \text{objective function} \tag{2.91}$$

Subject to:

$$g_j(X) \leq 0 \qquad j = 1, m \qquad \text{inequality constraints} \tag{2.92}$$

$$h_k(X) = 0 \qquad k = 1, l \qquad \text{equality constraints} \tag{2.93}$$

$$X_i^l \leq X_i \leq X_i^u \qquad i = 1, n \qquad \text{side constraints} \tag{2.94}$$

where

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix} \tag{2.95}$$

The design variables are defined as X_1 to X_n.

Most optimization algorithms start from an initial guess and are iterated upon to find the optimal solution. The most common form of the iterative procedure is

$$X^q = X^{q-1} + \beta^* S^q \tag{2.96}$$

where q is the iteration number, S is the search direction in the design space and β^* is the optimal step length in that direction .

2.2.1.1 Unconstrained Optimization. From calculus, we know that for F(X) to be a minimum, the gradient of F(X) must be equal to zero. This condition is necessary to say the point is a local minimum, but does not guarantee a global minimum. Also from calculus, we know that if the second derivative of the function with respect to all variables is positive definite, that point is a minimum. This leads to the definition of the Hessian matrix. The Hessian matrix is a matrix of second partial derivatives of the function with respect to the design variables. See Equation (2.97). If the Hessian is positive definite and the gradient is equal to zero, this ensures a *local minimum.* The design is a global minimum only if the Hessian is positive definite for all possible values of the design variables within the design space.

$$H = \begin{bmatrix} \frac{\partial^2 F(X)}{\partial X_1^2} & \frac{\partial^2 F(X)}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 F(X)}{\partial X_1 \partial X_n} \\ \frac{\partial^2 F(X)}{\partial X_2 \partial X_1} & \frac{\partial^2 F(X)}{\partial X_2^2} & \cdots & \frac{\partial^2 F(X)}{\partial X_2 \partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F(X)}{\partial Xn \partial X_1} & \frac{\partial^2 F(X)}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 F(X)}{\partial X_n^2} \end{bmatrix} \tag{2.97}$$

2.2.1.2 Constrained Optimization. For constrained optimization, it isn't necessary that the gradient of the objective vanish at the optimum. This type of optimization problem works to reduce the objective without violating the constraints. This is done by finding a search direction. A direction that will reduce the objective is called *usable.* A *feasible* direction is defined as one that will not violate a constraint if a small step is taken. An *active constraint* is defined as one that is within some tolerance of zero. A search direction is found that is feasible with respect to all active constraints. As stated, the search direction must be both usable and feasible.

$$\text{Usable Direction:} \qquad \nabla F(X) \cdot S \leq 0 \tag{2.98}$$

$$\text{Feasible Direction:} \quad \nabla g_j(X) \cdot S \leq 0 \quad \text{all j for which } g_j(X) = 0 \tag{2.99}$$

The optimum of a constrained optimization problem can be defined as meeting three criteria called the Kuhn-Tucker Necessary Conditions. These conditions are:

$$1.\ \mathbf{X}^* \text{ is feasible} \tag{2.100}$$

$$2.\ \lambda_j\, g_j(\mathbf{X}^*) = 0 \qquad j = 1, m \qquad \lambda_j \geq 0 \tag{2.101}$$

$$3.\ \nabla F(\mathbf{X}^*) + \ \textstyle\sum_{j=1}^{m} \lambda_j \ \nabla g_j(\mathbf{X}^*) + \ \displaystyle\sum_{k=1}^{l} \lambda_{k+m} \nabla h_k(\mathbf{X}^*) = 0 \tag{2.102}$$

$$\lambda_j \geq 0 \qquad \lambda_{k+m} \text{unrestricted}$$

where X^* is the vector of optimum design variables, g is the vector of inequality constraints, F is the objective, h is the vector of equality constraints and the λ's are Lagrange multipliers. The first Kuhn-Tucker condition (2.100) states that the optimal solution must satisfy all the constraints. The second Kuhn-Tucker condition (2.101) says that if the constraint is not equal to zero, then the Lagrange multiplier must be equal to zero. The last Kuhn-Tucker condition states that the objective gradient at optimum must be in the subspace spanned by the active constraint gradients. Also, for the Kuhn-Tucker conditions to be necessary for a constrained optimum, the X^* must be a *regular point*, (i.e., a design where the active constraint gradients are linearly independent). If the objective function and all constraint surfaces are convex, then the design space is said to be convex and the necessary Kuhn-Tucker conditions are also sufficient to guarantee that the optimum reached is a global optimum.

2.2.2 History of the Mixed Numerical Method. This section addresses the numerical approach to designing a mixed H_2/H_∞ controller. This method approaches an optimal fixed order solution. The controller from this design method will not satisfy the constraint exactly, but is as close as numerically possible within some tolerance.

Ridgely [Rid91] developed a numerical solution using a new performance index. This was based on a connection between the central H_∞ controller [DGKF89] and the mixed controller. This performance index is

$$J_\mu = (1-\mu)\|T_{zw}\|_2^2 + \mu\, tr[Q_\infty C_e^T C_e] \tag{2.103}$$

The value μ was varied from one to zero, resulting in a mixed controller. When μ equals 1, the central H_∞ controller is the solution. The DFP method was used in optimizing this set-up. There were several drawbacks to this method:

1. Amount of computational time required was on the order of one week for one point on the mixed H_2/H_∞ curve, since each μ design was optimized to a point on the mixed curve.

2. The order of the controller achieved by this method is the order of the mixed plant together with all of its weights.
3. This method does not allow for singular H_∞ constraints.

Walker developed a numerical approach which restated the mixed H_2/H_∞ optimization problem, where the two-norm is the objective and the infinity-norm is the constraint. The initial guess for the optimization process was the H_2 optimal controller, since it was a relatively easy point to calculate. Then steps in the infinity-norm constraint were made to develop the α^* versus γ^* curve. Inequality constraints were used to develop this performance index:

$$\min_{K\,stabilizing} \|T_{zw}(X)\|_2^2$$

subject to

$$\|T_{ed}(X)\|_\infty - \gamma \leq 0$$

This method saved time compared to Ridgely's method since each controller found was a point on the mixed curve. Walker initially used the DFP method to minimize the performance index. To prevent non-stabilizing controllers, the performance index was set to an artificially high value if an unstable closed-loop controller was calculated. Thus, that controller would not be accepted. Initially, equality constraints were addressed, but due to numerical drawbacks Walker then turned to using an inequality constraint approach. The final algorithm used to minimize the performance index was the MATLAB version of the SQP technique.

2.2.3 SQP Optimization.

The Sequential Quadratic Programming method uses a technique where the search direction is found by solving a quadratic objective function and linear constraints. The search direction is found by creating a quadratic approximation to the objective function. The general problem set-up is as follows:

$$\text{Minimize}: \quad Q(S) = F(X) + [\nabla F(X)]^T S + \tfrac{1}{2} S^T B S \tag{2.104}$$

$$\text{Subject to :} \qquad \nabla g_j(X) * S + \delta_j g_j(X) \leq 0 \qquad j = 1, m \tag{2.105}$$

$$\nabla h_k(X) * S + \bar{\delta} h_k(X) = 0 \qquad k = 1, l \tag{2.106}$$

The design variable for this subproblem is S, the search direction. The matrix B is updated through iterations to approach the Hessian matrix. The δ's ensure that the linearized constraints do not cut off the feasible region of the design space. These are problem dependent and should be chosen as close to 1 as possible.

$$\delta_j = 1 \quad \text{if } g_j(X) < 0 \tag{2.107}$$

$$\delta_j = \bar{\delta} \quad \text{if } g_j(X) \geq 0 \tag{2.108}$$

$$0 \leq \bar{\delta} \leq 1 \tag{2.109}$$

Once the search direction is found, a one dimensional search is performed with an exterior penalty function added.

$$\Phi = F(X) + \sum_{j=1}^{m} u_j \left\{ \max\left[0, g_j(X)\right] \right\} + \sum_{k=1}^{l} u_{m+k} \left[h_k(X) \right] \tag{2.110}$$

$$X = X^{q-1} + \beta S \tag{2.111}$$

$$u_j = |\lambda_j| \;\; j = 1, m + l \quad \text{first iteration} \tag{2.112}$$

$$u_j = \max\left[|\lambda_j|, \frac{1}{2}\left(u_j' + |\lambda_j| \right) \right] \quad \text{next iterations} \tag{2.113}$$

The variable u is a vector penalty parameter. The way this problem is set up, the one dimensional search is well conditioned and the step of $\beta = 1$ is a good step size to make once the matrix B approaches the Hessian matrix. Now that the first step has been taken, the B matrix needs to be updated. The B matrix improves the quadratic approximation for the search direction subproblem. There are several methods used to update B. The method used by MATLAB is the Broydon-Fletcher-Shanno-Goldfarb (BFGS) update formula [Van84].

$$B^* \;=\; B - \frac{B p p^T B}{p^T B p} + \frac{\eta \eta^T}{p \cdot \eta} \tag{2.114}$$

$$p = X^q - X^{q-1} \tag{2.115}$$

$$\eta = \theta y + (1-\theta)Bp \tag{2.116}$$

$$y = \nabla_x \phi(X^q, \lambda^q) - \nabla_x \phi(X^{q-1}, \lambda^{q-1}) \tag{2.117}$$

$$\phi(X,\lambda) = F(X) + \sum_{j=1}^{m} \lambda_j g_j(X) + \sum_{k=1}^{l} \lambda_{m+k} h_k(X) \tag{2.118}$$

$$\theta = \left\{ \begin{array}{c} 1.0 \text{ if } p \cdot y \geq 0.2 p^T B p \\ \frac{0.8 p^T Bp}{p^T Bp - p \cdot y} \text{ if } p \cdot y < 0.2 p^T Bp \end{array} \right\} \tag{2.119}$$

This update formula maintains positive definiteness and is called a variable metric method. This update method also allows for infeasible designs with no other additional methods required. Since the solution is allowed to approach from the infeasible design space, the resulting controller may be non-stabilizing. The previous performance index and problem set-up discussed in Section 2.2.2 can be used, but an additional stability constraint must be added.

2.2.4 Two-Norm Gradients. The gradient of the two-norm at H_2 optimal was defined by Ridgely [Rid91]. He used the definition of the two-norm

$$||T_{zw}||_2^2 = tr\left[Q_2 \mathcal{C}_z^T \mathcal{C}_z\right] \tag{2.120}$$

The H_2 problem can be set up as an optimization problem by forming a Lagrangian. This Lagrangian has the objective function as the definition of the two-norm. The constraint in the Lagrangian is the Lyapunov equation that must be solved to find a real, positive semidefinite solution for Q_2 in Equation (2.120). The Lagrangian is then

$$\mathcal{L} = tr\left[Q_2 \mathcal{C}_z^T \mathcal{C}_z\right] + tr\left[(\mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2^T + \mathcal{B}_w \mathcal{B}_w^T)\mathcal{X}\right] \tag{2.121}$$

Taking the partial derivatives of this equation with respect to all the unknown variables yields

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A_c} &= 2[X_{12}^T Q_{12} + X_2 Q_2] && (2.122)\\
\frac{\partial \mathcal{L}}{\partial B_c} &= 2[X_{12}^T Q_1 C_{y_2}^T + X_2 Q_{12}^T C_{y_2}^T + X_{12}^T V_{12}] && (2.123)\\
\frac{\partial \mathcal{L}}{\partial C_c} &= 2[B_{u_2}^T X_1 Q_{12} + B_{u_2}^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2] && (2.124)\\
\frac{\partial \mathcal{L}}{\partial Q_2} &= \mathcal{A}_2^T \mathcal{X} + \mathcal{X} \mathcal{A}_2 + \mathcal{C}_z^T \mathcal{C}_z && (2.125)\\
\frac{\partial \mathcal{L}}{\partial \mathcal{X}} &= \mathcal{A}_2 \mathcal{Q}_2 + \mathcal{Q}_2 \mathcal{A}_2 + \mathcal{B}_w \mathcal{B}_w^T && (2.126)
\end{aligned}$$

The process in which to calculate the H_2 gradient with respect to a specific controller is as follows

1. Set (2.125) and (2.126) equal to zero and solve for $\mathcal{X}$ and $\mathcal{Q}_2$
2. Compute $\frac{\partial \mathcal{L}}{\partial A_c}$, $\frac{\partial \mathcal{L}}{\partial B_c}$, and $\frac{\partial \mathcal{L}}{\partial C_c}$ from (2.122), (2.123), and (2.124), respectively
3. Rearrange the gradient by

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial Z} = \Big[& \left(\tfrac{\partial \mathcal{L}}{\partial A_c}\right)_1^T \ \cdots \ \left(\tfrac{\partial \mathcal{L}}{\partial A_c}\right)_{n_c}^T \ \left(\tfrac{\partial \mathcal{L}}{\partial B_c}\right)_1^T \ \cdots \\
& \left(\tfrac{\partial \mathcal{L}}{\partial B_c}\right)_p^T \ \left(\tfrac{\partial \mathcal{L}}{\partial C_c}\right)_1^T \ \cdots \ \left(\tfrac{\partial \mathcal{L}}{\partial C_c}\right)_m^T \Big]^T && (2.127)
\end{aligned}$$

A method to solve the derivative of the two-norm at a sub-optimal point was derived by Canfield [Can94]. He showed that the derivative of the two-norm can be derived through a *Direct Method.* The derivation begins with the definition of the two-norm.

$$\|T_{zw}\|_2^2 = tr[Q_2 C_z^T C_z] \tag{2.128}$$

where Q_2 is the solution to the following Lypaunov equation

$$A_2 Q_2 + Q_2 A_2^T + B_w B_w^T = 0 \tag{2.129}$$

The closed-loop matrices A_2, B_w and C_z are explicit functions of the controller matrices A_c, B_c, and C_c which are parameterized by the design vector P. Equation (2.128) can be differentiated with respect to the design vector P.

$$\|T_{zw}\|_2^{2'} = tr[Q_2{}'C_z^T C_z + QC_z^{T'}C_z + Q_2C_z^T C_z{}'] \tag{2.130}$$

Since Q_2 is an implicit function of P, the Lyapunov equation (2.129) can be differentiated with respect to each design variable. This is given by

$$A_2{}'Q_2 + A_2Q_2{}' + Q_2{}'A_2^T + Q_2A^{T'} + B_w{}'B_w^T + B_wB_w^{T'} = 0 \tag{2.131}$$

This can then be rearranged to form another Lyapunov equation:

$$A_2Q_2{}' + Q_2{}'A_2^T + (A_2{}'Q_2 + Q_2A^{T'} + B_w{}'B_w^T + B_wB_w^{T'}) = 0 \tag{2.132}$$

Then (2.132) can be solved for $Q_2{}'$ for each design variable. The derivative of the two-norm is then found by substituting $Q_2{}'$ into (2.130). This method was not used to calculate the H_2 gradients at sub-optimal mixed H_2/H_∞ designs since a Lyapunov equation would have to be solved for every design variable for every gradient. This would be too expensive. Rather, Canfield also derived an *Adjoint Method* for the gradient of the two-norm.

The *Adjoint Method* begins with the definition of the two-norm which will be called Ψ for this derivation.

$$\Psi(p, Q_2) = tr[Q_2C_z^T(p)C_z(p)] \tag{2.133}$$

Take the first variation of (2.133), treating p and Q_2 as independent variables.

$$\delta\Psi(p, Q_2) = \sum_i \frac{\partial\Psi(p, Q_2)}{\partial p_i}\delta p_i + \sum_j\sum_k \frac{\partial\Psi(p, Q_2)}{\partial Q_{2_{jk}}}\delta Q_{2_{jk}} \tag{2.134}$$

However, variations $\delta Q_{2_{jk}}$ are not independent of δp_i; rather, they are related through the Lyapunov equation (2.129). An identity can be formed by post-multiplying (2.129) by the adjoint matrix X and taking the trace.

$$\Upsilon(p, Q_2, X) = tr[(A_2 Q_2 + Q_2 A^T + B_w B_w^T)X] = 0 \tag{2.135}$$

The equation (2.135) will be called the adjoint equation. The variation of the adjoint equation is then

$$\frac{\partial \Upsilon(p, Q_2, X)}{\partial p_i}\delta p_i + \sum_j \sum_k \frac{\partial \Upsilon(p, Q_2, X)}{\partial Q_{2_{jk}}}\delta Q_{2_{jk}} = 0 \tag{2.136}$$

where X is not free to vary, but instead is defined as the matrix that satisfies

$$\frac{\partial \Upsilon(p, Q_2, X)}{\partial Q_2} = -\frac{\partial \Psi(p, Q_2)}{\partial Q_2} \tag{2.137}$$

which when combined with equation (2.133) and (2.135) can be written as

$$A_2^T X + X A_2 + C_z C_z^T = 0 \tag{2.138}$$

Therefore (2.138) is a Lyapunov equation for X that does not depend on any derivatives with respect to p (i.e., it is only solved once). Substituting equation (2.137) into (2.136) the following is found

$$\sum_j \sum_k \frac{\partial \Psi(p, Q_2, X)}{\partial Q_{2_{jk}}}\delta Q_{2_{jk}} = \sum_i \frac{\partial \Upsilon(p, Q_2, X)}{\partial p_i}\delta p_i \tag{2.139}$$

which provides the necessary relationship to determine how small variations δp of the design parameters cause corresponding small variations δQ_2 of the Lyapunov matrix Q_2. Next the total variation can be expressed by substituting equation (2.139) into (2.134).

$$\delta \Psi = \sum_i [\frac{\partial \Psi(p, Q_2)}{\partial p_i} + \frac{\partial \Upsilon(p, Q_2, X)}{\partial p_i}]\partial p_i \tag{2.140}$$

where the terms in brackets are recognized as the design derivatives that we seek

$$\frac{\partial \left\| T_{zw} \right\|_2^2}{\partial p} = \frac{\partial \Psi(p, Q_2)}{\partial p} + \frac{\partial \Upsilon(p, Q_2, X)}{\partial p} \tag{2.141}$$

where

$$\frac{\partial \Psi(p, Q_2)}{\partial p_i} = tr[Q_2 \frac{\partial C_z^T}{\partial p_i} C_z + Q_2 C_z^T \frac{\partial C_z}{\partial p_i}] \tag{2.142}$$

$$\frac{\partial \Upsilon(p, Q_2, X)}{\partial p_i} = tr[(\frac{\partial A_2}{\partial p_i} Q_2 + Q_2 \frac{\partial A_2^T}{\partial p_i} + \frac{\partial B_w}{\partial p_i} B_w^T + B_w \frac{\partial B_w^T}{\partial p_i})X] \tag{2.143}$$

and where Q_2 is found by solving the Lyapunov equation (2.129) and X is found by solving the Lyapunov equation (2.138).

Equation (2.143) represents the change in the two-norm with respect to the design variables in vector form. Equation (2.143) can be used to show the change in the two-norm with respect to the design variables in matrix format (i.e., A_c, B_c and C_c).

$$\frac{\partial \Psi(A_c, B_c, C_c, Q_2)}{\partial A_c} = 0 \tag{2.144}$$

$$\frac{\partial \Psi(A_c, B_c, C_c, Q_2)}{\partial B_c} = 0 \tag{2.145}$$

$$\begin{aligned} \frac{\partial \Psi(A_c, B_c, C_c, Q_2)}{\partial C_c} &= (Q_2 \begin{bmatrix} 0 & D_{zu} \end{bmatrix} C_z + \\ &\quad Q_2 C_z^T \begin{bmatrix} 0 & D_{zu} \end{bmatrix}) \end{aligned} \tag{2.146}$$

$$\frac{\partial \Upsilon(A_c, B_c, C_c, Q_2, X)}{\partial A_c} = (\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} Q_2 + Q_2 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix})X \tag{2.147}$$

$$\begin{aligned} \frac{\partial \Upsilon(A_c, B_c, C_c, Q_2, X)}{\partial B_c} &= (\begin{bmatrix} 0 & 0 \\ C_{y2} & 0 \end{bmatrix} Q_2 + Q_2 \begin{bmatrix} 0 & C_{y2} \\ 0 & 0 \end{bmatrix} + \\ &\quad \begin{bmatrix} 0 \\ D_{yw} \end{bmatrix} B_w^T + B_w \begin{bmatrix} 0 & D_{yw} \end{bmatrix})X \end{aligned} \tag{2.148}$$

$$\frac{\partial \Upsilon(A_c, B_c, C_c, Q_2, X)}{\partial C_c} = (\begin{bmatrix} 0 & B_{u2} \\ 0 & 0 \end{bmatrix} Q_2 + Q_2 \begin{bmatrix} 0 & 0 \\ B_{u2} & 0 \end{bmatrix})X \tag{2.149}$$

By simplifying Equations (2.144) through (2.149) and substituting them into (2.142) through (2.143) and then substituting the result into (2.141), in turn, the final equations for the derivative of the two-norm for a suboptimal mixed H_2/H_∞ design are found. The result can also be expressed in matrix form as

$$\frac{\partial\|T_{zw}\|_2^2}{\partial A_c} = 2[X_{12}^T Q_{12} + X_2 Q_2] \tag{2.150}$$

$$\frac{\partial\|T_{zw}\|_2^2}{\partial B_c} = 2[X_{12}^T Q_1 C_{y_2}^T + X_2 Q_{12}^T C_{y_2}^T + X_{12}^T V_{12}] \tag{2.151}$$

$$\frac{\partial\|T_{zw}\|_2^2}{\partial C_c} = 2[B_{u_2}^T X_1 Q_{12} + B_{u_2}^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2] \tag{2.152}$$

Therefore, the right side of (2.150) through (2.152) is the right side of (2.122) through (2.124).

The two-norm can be defined for both a stable and unstable closed-loop system. The two-norm is defined for an unstable transfer function as long as the system has no poles on the imaginary axis and is strictly proper. If the system is stable the two-norm is calculated by the methods discussed in Section 2.1.1. If the system is unstable, it can be broken into its stable and anti-stable parts. The two-norm of an anti-stable transfer function is the two-norm of its complex conjugate transpose, which is stable. Therefore, the two-norm of an antistable transfer function can be found by setting the state space matrices of the antistable system equal to $A = -A^T$, $B = C^T$ and $C = -B^T$. Then the two-norm can be calculated by the methods discussed in Section 2.1.1. Finally, through orthogonality, $\|G\|_2^2 = \|\Pi G\|_2^2 + \|\Pi^\perp G\|_2^2$ where ΠG is the stable projection and $\Pi^\perp G$ is the antistable projection. The two-norm and therefore the gradients must be calculated for stable and unstable closed-loop systems.

2.2.5 Infinity-Norm Gradients. The infinity-norm gradient can be calculated by an analytic method. This method is called a singular value sensitivity approach, which was developed by Geisy and Lim [GL93]. Assuming the maximum singular value of T_{ed} at X has a single peak, the derivatives of the infinity norm can be evaluated as:

$$\frac{\partial\|T_{ed}\|_\infty}{\partial x_i} = \Re e \left[u_1^H \left(\frac{\partial T_{ed}(\omega_0)}{\partial x_i} \right) \bigg|_{X_q} v_1 \right] \tag{2.153}$$

where u_1 and v_1 are the singular vectors associated with the maximum singular value of T_{ed}. The ω_o is the frequency at which the peak occurs, X_q is the current guess of X and $\Re e$ denotes the real part. The derivative of the transfer function with respect to the design variables can be written as

$$\left.\frac{\partial T_{ed}(\omega_0)}{\partial x_i}\right|_{X_q} = \left.\frac{\partial \left[\mathcal{C}_e\left(j\omega_0 - \mathcal{A}_\infty\right)^{-1}\mathcal{B}_d + \mathcal{D}_{ed}\right]}{\partial x_i}\right|_{X_q} \tag{2.154}$$

There are several advantages to this method over finite differencing:

1. The analytical method is quicker since there is only one gradient calculation per design variable. The finite difference method requires an infinity-norm calculation with the design variables perturbed forward and backward and then an average is taken between the two. The finite difference method therefore requires twice as many calculations as the analytical derivative.
2. The accuracy of the analytic gradient is better since in finite difference the solution is based on differences between very small numbers which can result in errors due to round off.
3. The H_∞ norm can be identified as a piecewise continuous function since the norm definition is the peak of the maximum singular value plot. The finite difference approach perturbs the controller and then finds the norm. The frequency where the peak is found could change and the gradient at a different frequency is usually not the same as the previous peak. Therefore, a central difference method could result in an incorrect gradient. The analytic method does not have this problem, since it provides information only on the unperturbed transfer function.

However, the sensitivity method has some disadvantages:

1. The method has difficulty converging when the peaks are located at more than one frequency. This results in a discontinuous gradient. This is a problem for both finite difference and the analytic gradient method.
2. The method depends on a search range over frequency. Previous knowledge of the shape of the singular value plot can be used to specify this range. A check needs to be made that the range given includes the maximum singular value.
3. The fineness of the frequency search can dictate the accuracy of the H_∞ norm.

III. Optimization Code Improvements

This section is divided into two parts. Section 3.1 discusses improvements made to the code. For each improvement the problem definition and solution are given. Section 3.2 discusses an implementation of the improvements for a simple SISO example.

3.1 Code Improvements

There were eight general improvements made to the mixed H_2/H_∞ optimization code. They are listed in order of importance relative to the improvement in run time and accuracy of the optimization code.

3.1.1 Peak Identification.

Problem Definition: The infinity-norm of T_{ed} is defined as the supremum over all frequency of the maximum singular value of T_{ed}. There can be several peaks in the singular value plot of a system. It was found that the optimization routine could not handle the case when two or more peaks had approximately the same singular value. The routine would constrain the peak at one frequency while the other frequency was not constrained. This caused the optimizer to switch back and forth between the peak values and not be able to complete the optimization iteration where the multiple peaks occurred. This is an identified limitation of using the analytic gradient method discussed by Geisy and Lim [GL93].

Problem Solution: The solution to this problem was to identify all peaks (and their frequencies) which were within a specified tolerance of the maximum. The peaks were identified by evaluating the change in slope of the singular value plot. The peaks were then separately constrained. The singular value at a frequency of zero rad/sec was automatically constrained and a frequency search range of 10^{-6} to 10^{4} rad/sec was used. A frequency band was placed around each peak to allow for movement of the peak as the controller design changed in the optimization routine. A function was implemented that allowed for a $\pm$ 0.3 rad/sec band about

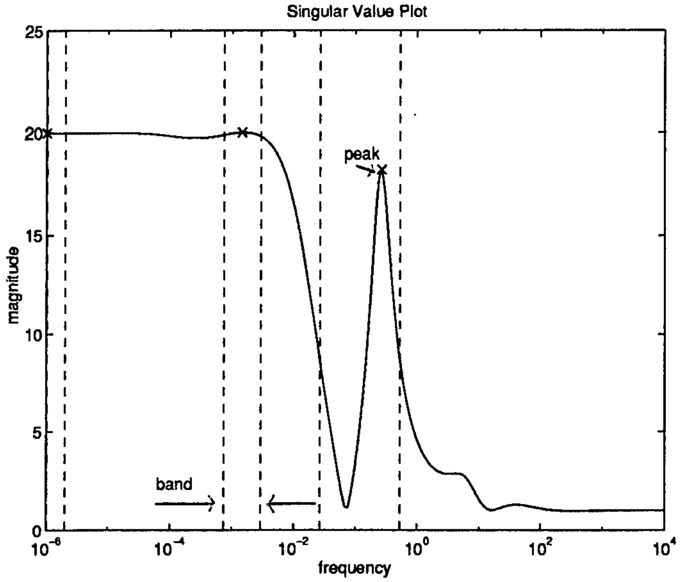

Figure 3.1 Singular Value Plot (γ=20)

the peak and also prevented the bands from overlapping. Two examples of singular value plots identifying peaks and bands are shown in Figures 3.1 and 3.2.

The frequency search for peak identification was done before and after each γ step. The search before the optimization was done to set up the bands once a peak was identified. The search after the optimization was to check that another peak had not risen above the constrained γ level outside the bands previously set. If a peak was identified outside the bands of the optimization, the bands from before the optimization and after the optimization would be combined. The bands were sorted and the frequencies were compared in log scale. If the difference between each band was less than 0.05 in log scale, then that band would be removed. This was a way to prevent overlapping and duplication of bands. Once the bands were reordered, the optimization sequence would begin again with the new bands and the last design vector found. The optimization sequence would not end until the H_∞ constraint was completely satisfied.

3.1.2 Gradient Calculation. The following five subsections will discuss the corrections made to the gradient calculations. They are listed in order of importance. First, the correction to the H_∞ gradient is presented and then the correction made to the H_2 gradient

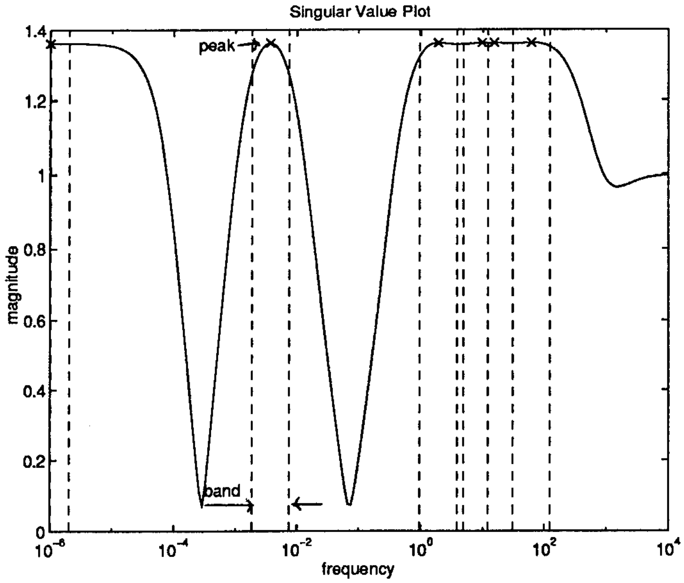

Figure 3.2 Singular Value Plot (γ=1.36)

follows. Finally, introduction of an analytic means to take the derivative of the stability constraint (previously done using finite differencing) is given.

3.1.2.1 H_∞ Gradient Modification.

Problem Definition: The analytic method for calculating the H_∞ gradient is faster, more accurate and better conditioned for finding a numeric answer than the finite difference method. However, when comparing the results of these two gradient methods, the gradients with respect to each design variable should be close in value. When comparing the H_∞ gradients, the results were significantly different. The gradients were opposite in sign in some cases. Another indication of a gradient problem was that the SQP optimization routine was regularly giving negative step sizes. This indicated the search direction the optimizer was calculating might be in error. This led to an investigation of the analytic gradient calculation for the infinity-norm.

Problem Solution: The background for the infinity-norm calculation is presented in Section 2.2.5. The two equations that define the analytic H_∞ gradient are

$$\frac{\partial \|T_{ed}\|_\infty}{\partial x_i} = \Re e \left[u_1^H \left(\frac{\partial T_{ed}(\omega_0)}{\partial x_i} \right) \Bigg|_{X_q} v_1 \right] \tag{3.1}$$

and

$$\left. \frac{\partial T_{ed}(\omega_0)}{\partial x_i} \right|_{X_q} = \left. \frac{\partial \left[\mathcal{C}_e \left(j\omega_0 - \mathcal{A}_\infty \right)^{-1} \mathcal{B}_d + \mathcal{D}_{ed} \right]}{\partial x_i} \right|_{X_q} \tag{3.2}$$

The important part of (3.2) is that it defines the derivative of the closed loop transfer function T_{ed} with respect to each value in the controller state space (i.e., the design variables for a numerical optimization solution). Using these two equations, the derivative of the infinity-norm of the closed loop T_{ed} transfer function can be found with respect to each design variable.

The original set-up of the mixed problem was to put the controller state space into modal form. Modal form places the real parts of the eigenvalues of the controller A matrix on the diagonal, with the imaginary parts of the eigenvalues on the superdiagonal and the negative of the imaginary parts of the eigenvalues on the subdiagonal. An example of a modal transformation is

$$A = \begin{bmatrix} 1 & 4 & 5 & 7 \\ 3 & 8 & 2 & 9 \\ 3 & 5 & 7 & 1 \\ 0 & 3 & 6 & 8 \end{bmatrix}$$

$$A_{modal} = \begin{bmatrix} 18.10 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.117 & 0.000 & 0.000 \\ 0.000 & 0.000 & 2.886 & 3.662 \\ 0.000 & 0.000 & -3.662 & 2.886 \end{bmatrix} \tag{3.3}$$

This modal form significantly reduces the amount of design variables needed to be found in an optimization routine. The higher the order of the controller, the more the savings in design variables. The reduction in number of design variables with the modal form comes from the fact that only the diagonal terms and superdiagonal terms in the A matrix need to be found. The subdiagonal terms do not need to be calculated since they are just the negate of the superdiagonal. The state space B and C matrices for the controller are assumed to be fully populated. An example of a fourth order controller A matrix is

$$A_{modal} = \begin{bmatrix} x_1 & x_5 & 0 & 0 \\ -x_5 & x_2 & x_6 & 0 \\ 0 & -x_6 & x_3 & x_7 \\ 0 & 0 & -x_7 & x_4 \end{bmatrix} \tag{3.4}$$

Notice for this example that putting the A matrix into modal form reduces the required number of design variables in A from 16 down to 7.

The derivative as defined in (3.2) must be taken with respect to every design variable. Previously, the gradients with respect to the subdiagonal terms were not factored into the overall gradient. This caused the gradients to be in error. This error caused examples of the search direction generated by the SQP optimization routine to be in opposite directions from what the finite difference gradient calculated. Adding the gradient with respect to the subdiagonal terms of the A matrix corrected this problem, and the finite difference matched very well with the analytic gradients for the infinity-norm.

3.1.2.2 H_2 Modification for Modal Form.

Problem Definition: The background of the H_2 gradient is given in Section 2.2.4. This development showed that the resulting gradients are a function of taking the partial derivative of the Lagrangian function with respect to each of the controller state space matrices. Since there is a partial with respect to the controller A matrix in the definition of the H_2 gradient, the subdiagonal gradient terms were neglected, similar to the H_∞ gradient problem. The following shows the Lagrangian equation and the partial of the Lagrangian with respect to A_c, (i.e., the controller A matrix):

$$\mathcal{L} = tr\left[Q_2\mathcal{C}_z^T\mathcal{C}_z\right] + tr\left[(\mathcal{A}_2 Q_2 + Q_2\mathcal{A}_2^T + \mathcal{B}_w\mathcal{B}_w^T)\mathcal{X}\right] \tag{3.5}$$

$$\frac{\partial\mathcal{L}}{\partial A_c} = 2(X_{12}^T Q_{12} + X_2 Q_2) \tag{3.6}$$

The problem with the H_2 gradients was identified in the same way as the H_∞ gradients. The H_2 gradients were significantly in error from finite difference without the subdiagonal terms.

Problem Solution: Adding the derivative with respect to the subdiagonal terms to the equation for the partial of the Lagrangian with respect to A_c corrected this problem.

3.1.2.3 H_2 Modification for Sub-Optimal Design.

Problem Definition: Comparison of the finite difference to the analytical two-norm gradients for each design variable showed a difference of a factor of two. This lead to a further investigation of the analytic two-norm gradient. Through calculation of the necessary conditions it was found that there was a factor of two in each of these derivatives. This factor was ignored in the two-norm gradient equations, since this calculation was considered at the H_2 optimal point and therefore the necessary conditions would be equal to zero.

Problem Solution: If the mixed design is not at the H_2 optimal point, the mixed design is by definition sub-optimal. Therefore, the gradients of the Lagrangian of H_2, when completing a mixed H_2/H_∞ optimization design, should contain the factor of two for each derivative with respect to the controller state space matrices. The two-norm derivative at sub-optimal designs can be written as

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A_c} &= 2 * [X_{12}^T Q_{12} + X_2 Q_2] && (3.7)\\
\frac{\partial \mathcal{L}}{\partial B_c} &= 2 * [X_{12}^T Q_1 C_{y_2}^T + X_2 Q_{12}^T C_{y_2}^T + X_{12}^T V_{12}] && (3.8)\\
\frac{\partial \mathcal{L}}{\partial C_c} &= 2 * [B_{u_2}^T X_1 Q_{12} + B_{u_2}^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2] && (3.9)\\
\text{Solve for } \mathcal{X} : & \quad [\mathcal{A}_2^T \mathcal{X} + \mathcal{X} \mathcal{A}_2 + \mathcal{C}_z^T \mathcal{C}_z] = 0 && (3.10)\\
\text{Solve for } Q_2 : & \quad [\mathcal{A}_2 Q_2 + Q_2 \mathcal{A}_2 + \mathcal{B}_w \mathcal{B}_w^T] = 0 && (3.11)
\end{aligned}
$$

3.1.2.4 H_2 Gradient Modification if Unstable.

Problem Definition: The SQP optimization method is allowed to begin with an initial guess that is in the infeasible region. SQP also allows a feasible solution to go into the infeasible region during the optimization process. A non-stabilizing controller is defined when the stability constraint is violated, and therefore the design is in the infeasible region. Since it is possible that the controller could be non-stabilizing, the code should be able to calculate the two-norm for this case. The previous code did a stable/anti-stable projection to calculate the two-norm, but did not consider this projection for the gradient of the two-norm.

Problem Solution: An addition was made to the code to calculate the H_2 gradient when the controller was identified as non-stabilizing. This gradient was calculated using finite differencing. The section of the code that calculates the gradient using finite differencing is called whenever the real part of the maximum eigenvalue of the closed loop A_2 matrix of the H_2 problem is positive.

3.1.2.5 Stability Gradient Modification.

Problem Definition: The previous method calculated the stability gradient using the finite difference method. This method became costly due to the fact that for every design variable, two eigenvalue evaluations must be done. This must be done for every derivative calculation in the gradient. A method was derived to find the gradients of the stability constraint analytically.

Problem Solution: The stability constraint is defined as requiring the real part of the maximum eigenvalue of the closed-loop A_2 matrix to be negative. The derivative of this constraint is the derivative of the eigenvalue. The eigenvalue of the closed-loop A_2 matrix can be expressed as

$$\lambda = u_2^H A_2 v_2 \tag{3.12}$$

where u_2 and v_2 are the left and right eigenvectors, respectively, associated with the eigenvalue λ. The stability constraint is

$$\max \Re e(\lambda) \leq 0 \tag{3.13}$$

so the stability constraint gradient is

$$\Re e \frac{\partial \lambda}{\partial x_i} = \Re e[u_2^H \frac{\partial A_2}{\partial x_i} v_2] \tag{3.14}$$

The mixed H_2/H_∞ A_2 matrix is an explicit function of the design variables. Taking the derivative of A_2 with respect to each design variable, and then pre- and post- multiplying it by the normalized eigenvectors produced the analytic stability gradient. This allowed one eigenvalue evaluation per stability gradient instead of 2n+1 evaluations, where n=number of design variables. Although eigenvalue calculations are not expensive, the cost increases with an increase in the number of design variables. Note that this development assumes that the maximum eigenvalue is not repeated. If it were a repeated eigenvalue, further modifications would need to be made. These were not considered here.

3.1.3 Penalty Function.

Problem Definition: SQP allows infeasible designs in the optimization process; therefore, a design (i.e., controller) may be non-stabilizing. If the controller is non-stabilizing, the calculation of the two-norm gradient may be very expensive. The expense is due to the finite differencing routine that was added to calculate the two-norm gradient when the stability constraint was violated. Therefore, it would be advantageous to prevent non-stabilizing controller designs while in the optimization process.

Problem Solution: The solution to this problem was to add a penalty to the objective function which was a function of the stability constraint. The penalty function ϕ added is

$$\phi = [\lambda_3 * \text{ max(real(eig(}A_2\text{)))}]^2 \tag{3.15}$$

where λ_3 is the scale factor used to prevent unstable designs. This will be discussed in the scaling section.

If the controller is non-stabilizing, the gradient of the two-norm must also have the penalty function gradient added. The derivative of the penalty function is shown in (3.16). This equation was added to the original two-norm gradient when the stability constraint was violated. Therefore,

$$\frac{\partial\phi}{\partial x_i} = 2 * \lambda_3^2 * \text{max(real(eig(}A_2\text{)))} * (\frac{\partial\lambda}{\partial x_i}) \tag{3.16}$$

where $\frac{\partial\lambda}{\partial x_i}$ is defined in Section 3.1.2.5.

3.1.4 Use of FMIN for H_∞ Norm Calculation.

Problem Definition: The highest percentage of computer time used to calculate the H_2/H_∞ curve was spent doing the frequency search over the singular value plot. Since the infinity-norm is found by a frequency search over the singular value plot for the maximum magnitude, the finer the search grid over this plot the better the infinity-norm calculation. The finer the grid, however, the more expensive the calculation of the H_∞ norm.

Problem Solution: Use of a function in MATLAB called FMIN to find the maximum point within defined bands for each peak improves the run time. A frequency search is still required to identify the peaks and set up the bands for the problem, but this search is only done one time per γ step. The original code did a frequency search for every function evaluation. The frequency search for the new code can also use many fewer points since it only needs slope information to identify the peaks and generate the bands, but is no longer required to find the precise value of the infinity-norm.

Each function evaluation requested by the SQP optimization routine solves for the H_∞ norm for each band using FMIN. This function finds the minimum of an unconstrained function of one variable within a fixed interval. The function that is to be minimized must be continuous. Also, FMIN may return a local minimum if the function it is given is not unimodal. This method was used to find the minimum of the negative of the singular value plot since FMIN finds the minimum of a function. It was found that using FMIN provided significantly fewer evaluations and took less time to run than the previous frequency search. An example of the savings is where previously 800 points were used to search over the singular value plot to find the maximum value, FMIN would find the maximum in approximately 20 function evaluations. This equates to a 97 percent savings in time. This comparison was made when there was only one peak in the singular value plot, and therefore only one band. FMIN also offers the advantage of specifying the tolerance to which the infinity-norm is calculated without a significant increase in the number of function evaluations.

3.1.5 Scaling.

Problem Definition: Numerical difficulties are often encountered in an optimization process when one constraint is larger or changes more quickly than another constraint. Difficulties are also encountered when the value of the objective is not of the same order of magnitude as the constraints. The goal then is to normalize the objective and constraints to unity before each optimization run (i.e., for each γ step for the mixed H_2/H_∞ problem).

The original code for this calculation used scale factors to multiply the objective and constraints. These scale factors were arbitrarily set, depending on the problem and the point on the H_2/H_∞ curve where the optimization is begun.

Problem Solution: The H_2 objective was normalized by the last value of the two-norm. A constant scale factor was used to multiply the normalized H_2 objective. See (3.17). It was found that by scaling the objective constantly in this manner, the time spent in the line search was much less. A scale factor of (0.001) was used for all examples shown in this thesis. The scale factor used for the infinity-norm was found by normalizing the infinity-norm constraint using the current set constraint value of γ. See (3.18). Finally, a constant scale factor of (1000) was used for the stability constraint. See (3.19). The stability constraint was not normalized. The intent for this constraint was that it always stay negative, not that it approach zero. This scale factor, along with the penalty function discussed earlier, worked well to prevent a non-stabilizing controller design. The scalings were

$$\begin{aligned}\text{Scalle for 2-Norm Objective} &= 1/\|T_{zw_{q-1}}\|_2^2 * (1/1000) && (3.17)\\ \text{Scale for Infinity-Norm Constraint} &= 1/\|T_{ed_{q-1}}\|_\infty && (3.18)\\ \text{Scale for Stability Constraint} &= 1000 && (3.19)\end{aligned}$$

where $\|T_{zw_{q-1}}\|_2^2$ and $\|T_{ed_{q-1}}\|_\infty^2$ are the two-norm calculation and the constraint level set for the infinity-norm for the previous γ iteration. The objective and constraints then become

$$\begin{aligned}\text{Objective} &= 1/\|T_{zw_{q-1}}\|_2^2 * (1/1000)\|T_{zw}\|_2^2 && (3.20)\\ \text{Infinity-Norm Constraint} &= 1/\|T_{ed_{q-1}}\|_\infty \|T_{ed}\|_\infty && (3.21)\\ \text{Stability Constraint} &= 1000[\max \Re e(\lambda)] \leq 0 && (3.22)\end{aligned}$$

3.1.6 Modifications for MIMO Designs.

Problem Definition: The code for mixed H_2/H_∞ optimization needs to be setup to calculate a solution for a MIMO problem. The infinity-norm calculation must search multiple singular

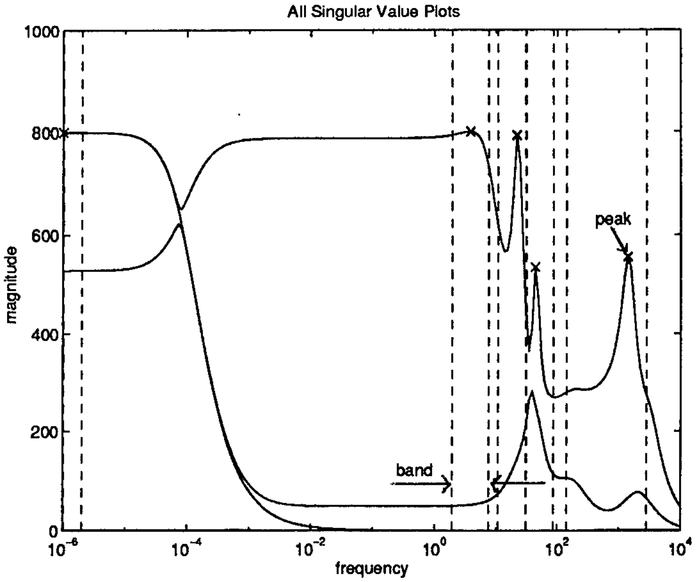

Figure 3.3 Singular Value Plot for MIMO Example (γ=800)

value plots to find a maximum and therefore the calculation of peaks and bands must be adjusted.

Problem Solution: The modification that needed to be made for this case was for the H_∞ norm calculation. The maximum singular value over all the singular value plots for the MIMO system was found. The FMIN function was still used to calculate the peak values within the bands. Two examples of MIMO singular value plots with bands and peaks identified are shown in Figures 3.3 and 3.4.

3.1.7 Automatic H_∞ Steps.

Problem Definition: The improvements to the code significantly reduced the run time required to generate the H_2/H_∞ curve. However, large step sizes in γ caused numerical ill conditioning of the problem. This caused the user to set a step size, run the optimization code, stop the code when ill conditioning occurred, save the controller and restart the optimizer with a different step size.

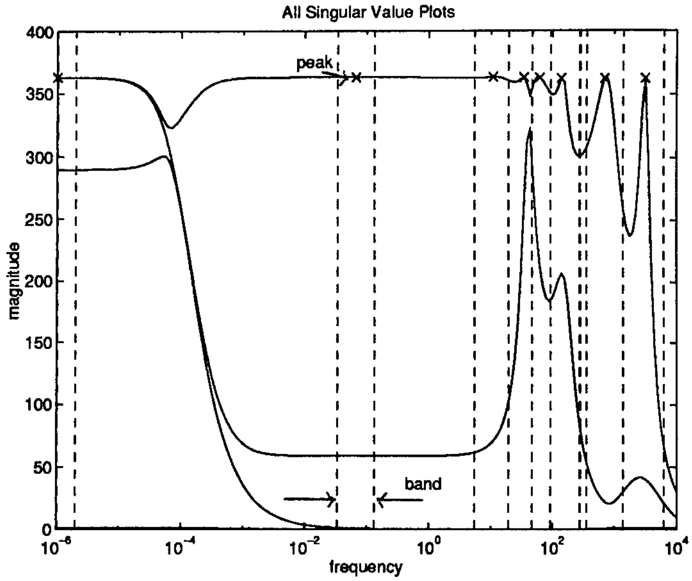

Figure 3.4 Singular Value Plot for MIMO Example (γ=360)

Problem Solution: A routine was added that would automatically set the step size, and therefore generate the entire mixed curve automatically. The *autostep* function begins by taking two γ steps from H_2 optimal. The information from these two steps is then used to form a linear approximation of the mixed curve. An estimate is made using that approximation of what size the next γ step should be for a specified change in the two-norm. See Figure 3.5. If the step size is larger than the original step size it is ignored. If the step size is smaller than the original γ step size, the next iteration is set at that γ level. The original specified levels of percentage changes of infinity-norm and two-norm can be input by the designer.

This automatic routine works well when the designer chooses a small percentage change in γ for the first step size. The larger the percentage the better the chance of numerical ill conditioning. The examples run using this routine used a change in α of 5 percent and a change in γ of 5 percent.

3.1.8 Requiring H_∞ Constraint to be Active.

Problem Definition: A theorem was given by Walker [Wal94] which proved that if the controller order is greater than or equal to the order of the H_2 problem, then the mixed H_2/H_∞

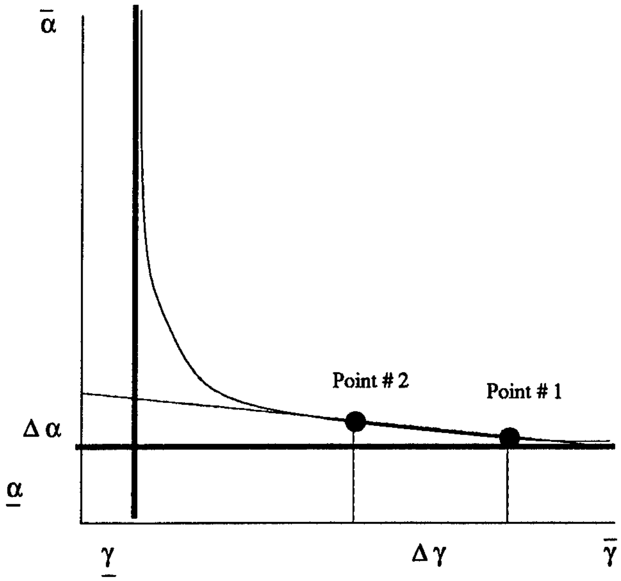

Figure 3.5 Linear Projection to Estimate Gamma Step

solution lies on the boundary of the H_∞ constraint. It was found that by setting the optimization problem up with an inequality constraint occasionally would result in cases where the H_∞ constraint would not be active. If the problem was set up with equality constraints, the optimizer would try to force all the H_∞ constraints from the frequency bands to zero. However, when multiple H_∞ peaks were present, it is not possible to know in advance which peak should be constrained as an equality.

Problem Solution: The mixed H_2/H_∞ active constraint requirement was that only the maximum H_∞ constraint (from the bands) must be active. Therefore, a modification was made to the MATLAB SQP routine. The modification required that the maximum H_∞ constraint was close to zero within some tolerance. This change resulted in the H_∞ constraint being active at each step along the mixed H_2/H_∞ curve. It should be noted that this requirement is only true for one H_∞ constraint. If multiple H_∞ constraints are added to the problem, (i.e., $||T_{ed1}||_\infty$ and $||T_{ed2}||_\infty$, not constraints from several peaks of one H_∞ problem), then only one H_∞ constraint is required to be active. Therefore, in the multiple H_∞ constraint case, this active requirement is dropped.

3.2 Numerical Test Example

A SISO example was chosen to demonstrate these fixes. It does not represent any physical system. The example is taken from [DS79], where it was used to demonstrate how to recover robustness with observers. This study used this example to demonstrate the fixes to the optimization code because of the limited number of design variables required to develop the controller. It was also chosen because it had a low γ at H_2 optimal, and would require less optimization time to generate the mixed H_2/H_∞ curve. This was important for ease of making comparisons between the previous and the improved numerical method.

3.2.1 Problem Set-Up. The equations for the system can be represented as

$$\begin{aligned}
\dot{x} &= Ax + Bu + \Gamma\xi && (3.23)\\
&= \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 35 \\ -61 \end{bmatrix} \xi && (3.24)\\
y &= Cx + n && (3.25)\\
&= [2 \quad 1]\, x + n && (3.26)
\end{aligned}$$

where the open-loop plant transfer function for this system is given by

$$P_{yu} = \frac{s+2}{(s+1)(s+3)} \tag{3.27}$$

3.2.1.1 H_2 Problem. The objective of the H_2 design is to find an internally stabilizing controller which minimizes the response of the system when disturbances and noises are present. For this example, the H_2 problem is a standard LQG problem. The following matrices were chosen in constructing the regulator [DS79]:

$$\begin{aligned}
z &= Hx = 4\sqrt{5}\,[\sqrt{35} \quad 1]\, x && (3.28)\\
Q_c &= H^T H = 80 \begin{bmatrix} 35 & \sqrt{35} \\ \sqrt{35} & 1 \end{bmatrix} && (3.29)
\end{aligned}$$

$$R_c \;=\; 1 \tag{3.30}$$

These choices produce an optimal control law of

$$u = -K_c x = -\,[\,50 \quad 10\,]\,x \tag{3.31}$$

which produces the closed-loop regulator poles

$$\lambda_{cl} = -7.0 \pm j2.0 \tag{3.32}$$

An estimator (i.e., Kalman Filter) was designed to reconstruct the states. The following matrices were selected for the Kalman Filter design [DS79]:

$$Q_f \;=\; \Gamma Q_o \Gamma^T = \begin{bmatrix} 35 \\ -61 \end{bmatrix} (1)\,[\,35 \quad -61\,] \tag{3.33}$$

$$R_f \;=1 \tag{3.34}$$

The resulting H_2 matrices are as follows:

$$A_2 \;=\; \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \tag{3.35}$$

$$B_w \;=\; \begin{bmatrix} 35 & 0 \\ -61 & 0 \end{bmatrix} \tag{3.36}$$

$$B_{u2} \;=\; \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.37}$$

$$C_z \;=\; 4 * \sqrt{5} \begin{bmatrix} \sqrt{35} & 1 \\ 0 & 0 \end{bmatrix} \tag{3.38}$$

$$C_{y2} \;=\; [\,2 \quad 1\,] \tag{3.39}$$

$$D_{zw} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.40}$$

$$D_{yw} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{3.41}$$

$$D_{yu} = [0] \tag{3.42}$$

3.2.1.2 H_∞ Problem. The H_∞ problem is a weighted sensitivity minimization problem. It is used to improve tracking performance. The closed loop sensitivity was chosen to give good noise rejection, tracking at low frequency and attenuate system response to high frequency noise and uncertainties. The H_∞ problem can be set-up to minimize the weighted sensitivity of a system. The sensitivity weight was selected as the inverse of the desired sensitivity, given by

$$W_s = \frac{s + 12.5}{s + 0.7} \tag{3.43}$$

The state space representation of the weight can be written as

$$A_s = [-0.7] \tag{3.44}$$

$$B_s = [1] \tag{3.45}$$

$$C_s = [11.0] \tag{3.46}$$

$$D_s = [1] \tag{3.47}$$

The resulting H_∞ matrices are as follows:

$$A_\infty = \begin{bmatrix} A_2 & 0 \\ B_s * C_{y2} & A_s \end{bmatrix} \tag{3.48}$$

$$B_d = \begin{bmatrix} 0 \\ B_s \end{bmatrix} \tag{3.49}$$

$$B_{u\infty} = \begin{bmatrix} B_{u2} \\ 0 \end{bmatrix} \qquad (3.50)$$

$$C_e = \begin{bmatrix} D_s * C_{y2} & C_s \end{bmatrix} \qquad (3.51)$$

$$C_{y\infty} = \begin{bmatrix} C_{y2} & 0 \end{bmatrix} \qquad (3.52)$$

$$D_{ed} = [D_s] \qquad (3.53)$$

$$D_{eu} = [0] \qquad (3.54)$$

$$D_{yd} = [1] \qquad (3.55)$$

$$D_{yu} = [0] \qquad (3.56)$$

where the submatrices above were given previously.

3.2.2 Test Example Results. The numerical code, as it was prior to this work, was run for this example. The full mixed H_2/H_∞ curve could not be generated. Due to numerical problems discussed earlier in this chapter, the lowest γ level that could be achieved was γ=4. The $\underline{\gamma}$ for this problem was found using a routine in MATLAB called HINFSYN. The minimum γ achievable for the third order H_∞ problem was found to be $\underline{\gamma}$=1.0. The mixed H_2/H_∞ controller was run at the order of the H_2 problem, which is 2. Thus, $\underline{\gamma}$=1.0 is not necessarily achievable at order 2, but as will be seen, it can be reduced to much lower than 4.

The problem of the multiple singular value peaks was found to be the reason the solution close to $\underline{\gamma}$ could not be reached. When the frequency search was divided into two regions, (see Figure 3.6), and separate H_∞ constraints were set for each region, γ=1.001 was achieved. A comparison of the mixed curves before the division for the peaks and after is shown in Figure 3.7 and Figure 3.8. The step size in γ was unity.

The solution time using the method of manually dividing the peaks and running the original code was approximately 14 hours. This is an excessive amount of time when considering that this was a second order control problem that only required seven design variables and had a maximum of two peaks (i.e., two H_∞ constraints). Also, the number of controllers generated in this time frame was only 15. Finally, this method required a manual division

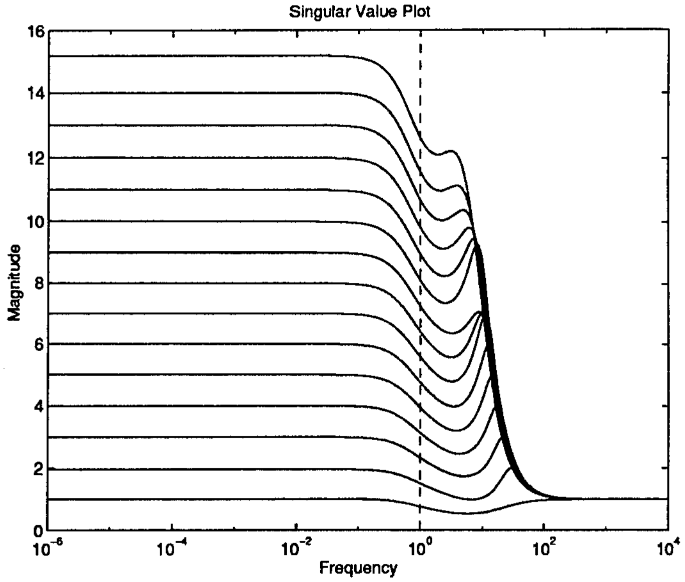

Figure 3.6 Singular Value Plot for Test Example

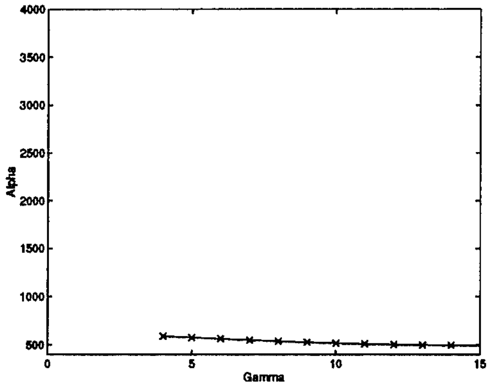

Figure 3.7 Mixed H_2/H_∞ Curve Original Code

of peaks by the designer. The division of peaks was relatively simple for this problem since there were only two peaks, but for problems that had four or five peaks this manual division technique would be very time consuming.

Another comparison between the old and new method is the maximum number of function and gradient evaluations required (see Table 3.1). The old method, when the peaks were divided manually, required an average number of function evaluations of approximately 400 per γ iteration. The new method required only an average of approximately 10 function evaluations per γ iteration.

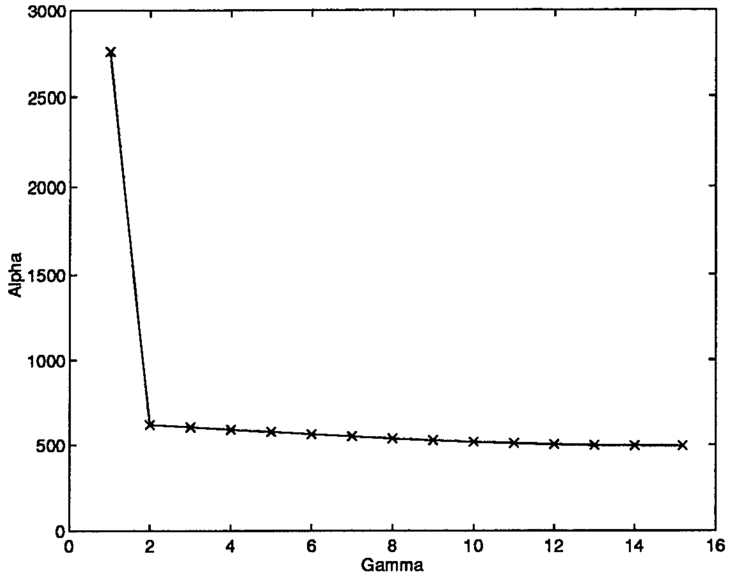

Figure 3.8 Mixed H_2/H_∞ Curve Improved Code

Table 3.1 Number of Function and Gradient Evaluations Comparison

	Original Code		New Code	
γ	NUM FUN	NUM GRAD	NUM FUN	NUM GRAD
14	88	32	4	4
13	285	40	4	4
12	685	65	4	4
11	344	76	4	4
10	92	25	4	4
9	685	64	11	11
8	689	57	4	4
7	10	10	5	5
6	11	11	5	5
5	11	11	5	5
4	11	11	5	5
3	688	58	6	6
2	12	12	6	6
1	694	231	119	98
TOTAL	4305	703	186	165

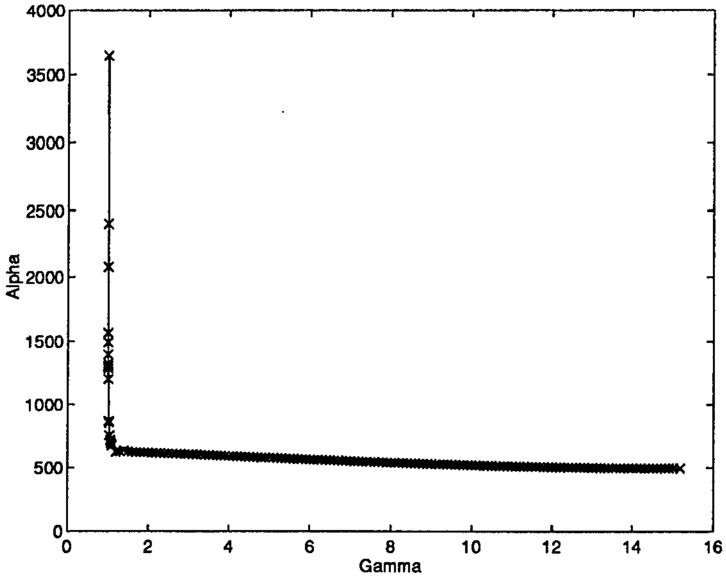

Figure 3.9 Mixed H_2/H_∞ curve with over 150 controllers

Note that the maximum number of function evaluations per γ step was set at 700. There were five cases in the original code where number of function evaluations was close to 700; in these cases, the optimizer was stopped at maximum iterations. The value of the size of the line search step for each of these cases was consistently negative, indicating the solution to within the specified tolerance would not have been found. Controllers with infinity-norms very close to tolerance were generated in this manner.

Using the improvements discussed in Section 3.1, the run time was reduced from 14 hours down to 2 minutes as measured in CPU time. The CPU time compared very closely to real time when working on a dedicated SPARC20 SUN workstation. Many more controllers could also be calculated. The time required to calculate the 150 controllers shown in Figure 3.9 was 17 CPU minutes.

3.3 Summary

This chapter presented eight general improvements made to the optimization code for the mixed H_2/H_∞ controller design. It discussed how each problem was recognized and a solution implemented. These improvements were then tested on a small example to compare improved efficiency. The example shows that a solution for the mixed problem could be found very close to $\underline{\gamma}$, where previously this controller could not be found. The mixed H_2/H_∞ controller came very close to the absolute minmum $\underline{\gamma}$, but could not equal it since it was computed at an order less than the order of the H_∞ problem. The example also showed that the new code ran approximately 200 times faster than the previous code. This was calculated for the example where the frequency search was manually divided and a complete mixed curve could be generated. Finally, this example demonstrates the speed in which many controllers along the mixed H_2/H_∞ curve could be found. The results showed it took 17 CPU minutes for 150 controllers to be calculated with the new code and 14 CPU hours for 15 controllers to be calculated with the old code and the manually divided peaks.

IV. F-16 Example

4.1 Introduction

The purpose of this chapter is to design mixed H_2/H_∞ controllers for a normal acceleration command following model of the AFTI F-16. This example was first examined by Luke [Luk93]. He designed both LQG/LTR and mixed H_2/H_∞ controllers for this problem. Luke's study was done to compare the performance of the two controllers and draw conclusions about the effectiveness of the mixed H_2/H_∞ design. After completion, it was found that the bandwidth for the LTR design was different than the bandwidth for the sensitivity weighting of the H_∞ design. The maximum bandwidth for the LQG/LTR design was 10 rad/sec, whereas the H_∞ sensitivity weighting's bandwidth was 1 rad/sec. The objective for this thesis was to change the H_∞ sensitivity weighting's bandwidth to 10 rad/sec, rerun the mixed H_2/H_∞ controllers, and compare them to the LQG/LTR design calculated by Luke.

Another objective for this example is to examine controllers of orders different than that of the H_2 problem. Three different order controllers were evaluated. The first was at the order of the H_2 problem and the second was the order of the H_∞ problem. Finally, a reduced order mixed H_2/H_∞ controller, which was one order lower than the H_2 problem, was also evaluated.

This chapter first defines the F-16 problem set-up in Section 4.2. The mixed H_2/H_∞ controllers for different orders are defined and analyzed in Section 4.3. Finally, the comparisons between the LQG/LTR and the mixed H_2/H_∞ designs are presented in Section 4.4.

4.2 Problem Set-Up

This example is a normal acceleration command following model of an AFTI F-16 flying at sea level, Mach 0.6. The linear model consists of a servo, longitudinal dynamics and a delay. This is shown in Figure 4.1. A more detailed model of the F-16 model is shown in Figure 4.2 This section will first discuss the state space development for the H_2 and H_∞

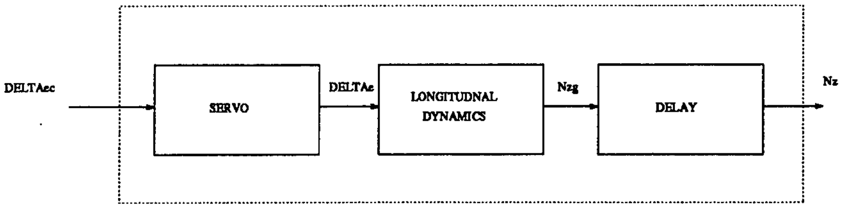

Figure 4.1 F-16 Plant

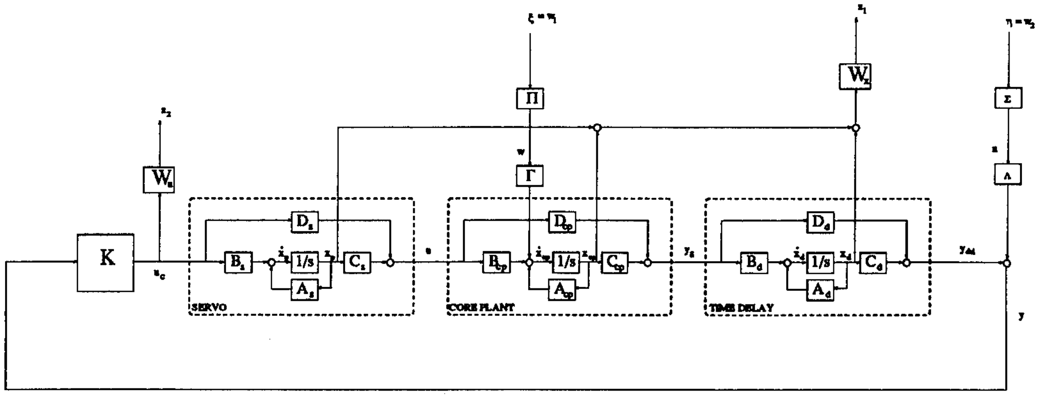

Figure 4.2 F-16 Plant

problems. Next, the noise model used for this example will be developed. Finally, a brief description of an LQG/LTR design will be presented.

The states for the core plant are

$$x_{cp} = \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} \tag{4.1}$$

In (4.1), the states, u, α, θ and, q are perturbations in the forward speed, angle of attack, pitch angle, and pitch rate, respectively. All angles and angular rates are in *radians* and *radians/sec* and velocity is in *ft/sec*.

4.2.1 Servo. The servo is modeled as a first order lag with a pole at -20. This is represented in transfer function form as

$$\frac{\delta_e}{\delta_{ec}} = \frac{20}{s+20} \tag{4.2}$$

This can be written in state space format as

$$\dot{x}_{\delta_e} = [-20]x_{\delta_e} + [20]\delta_{e_c} \tag{4.3}$$

$$\delta_e = [1]x_{\delta_e} + [0]\delta_{e_c} \tag{4.4}$$

4.2.2 Longitudinal Dynamics. The longitudinal dynamics of the F-16 at sea level flying at Mach 0.6 were obtained from Baird [Bai92].

$$\dot{x}_{cp} = \begin{bmatrix} -0.0015 & 37.382 & -32.200 & -17.940 \\ -0.0001 & -1.4910 & -0.0013 & 0.9960 \\ 0 & 0 & 0 & 1.0000 \\ -0.0004 & 9.7530 & 0.0003 & -0.9600 \end{bmatrix} x_{cp} + \begin{bmatrix} 0.0021 \\ -0.1880 \\ 0 \\ -19.0400 \end{bmatrix} \delta_e \tag{4.5}$$

$$N_{z_g} = \begin{bmatrix} 0.0015 & 35.2640 & 0.0272 & -0.3340 \end{bmatrix} x_{cp} + [-4.3660]\delta_e \tag{4.6}$$

4.2.3 Time Delay. For the time delay, a period of 0.05 seconds is chosen, which leads to a first order Padé approximation transfer function of

$$\frac{N_z}{N_{zg}} = \frac{40-s}{s+40} \tag{4.7}$$

The corresponding state space representation of the delay is

$$\dot{x}_d = [-40]x_d + [1]N_{z_g} \tag{4.8}$$

$$N_z = [80]x_d + [-1]N_{z_g} \tag{4.9}$$

4.2.4 Complete State Space. The servo, longitudinal dynamics, and delay can be combined into one complete state space. The states x_{cp} represent those for the core plant (the longitudinal dynamics). The states x_{δ_e} represents those for the servo, and the states x_d represent those for the time delay.

$$\begin{bmatrix} \dot{x}_{cp} \\ \dot{x}_{\delta_e} \\ \dot{x}_d \end{bmatrix} = \begin{bmatrix} A_{cp} & B_{cp}C_{\delta_e} & 0 \\ 0 & A_{\delta_e} & 0 \\ B_dC_{cp} & B_dD_{cp}C_{\delta_e} & A_d \end{bmatrix} \begin{bmatrix} x_{cp} \\ x_{\delta_e} \\ x_d \end{bmatrix} + \begin{bmatrix} B_{cp}D_{\delta_e} \\ B_{\delta_e} \\ B_dD_{cp}D_{\delta_e} \end{bmatrix} \delta_{e_c} \tag{4.10}$$

$$N_{z_{del}} = \begin{bmatrix} D_dC_{cp} & D_dD_{cp}C_{\delta_e} & C_d \end{bmatrix} \begin{bmatrix} x_{cp} \\ x_{\delta_e} \\ x_d \end{bmatrix} + [D_dD_{cp}D_{\delta_e}]\delta_{e_c} \tag{4.11}$$

or, substituting in the values,

$$\dot{x}_p = \begin{bmatrix} -0.0015 & 37.3820 & -32.2000 & -17.9400 & 0.0021 & 0 \\ -0.0001 & -1.4910 & -0.0013 & 0.9960 & -0.1880 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ -0.0004 & 9.7530 & 0.0003 & -0.9600 & -19.040 & 0 \\ 0 & 0 & 0 & 0 & -20 & 0 \\ 0.0015 & 35.2640 & 0.0272 & -0.3340 & -4.366 & -40.000 \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 20 \\ 0 \end{bmatrix} \delta_{e_c} \tag{4.12}$$

$$N_{z_{del}} = \begin{bmatrix} -0.0015 & -35.264 & -0.0272 & 0.3340 & 4.3360 & 80.000 \end{bmatrix} x_p + [0]\delta_{e_c} \tag{4.13}$$

4.2.5 Noise. The noise models for the LQG/LTR design are given in this subsection. These noises were used in all simulations. The noises introduced in this system were wind gusts and sensor noise. The wind gusts were introduced as perturbations in angle of attack. They were modeled as UWGN inputs that were then filtered. The sensor noise, which entered at the output of the plant, was also modeled as a filtered UWGN.

The filter used for the wind gust was developed by Baird [Bai92]. It is a low pass filter which cuts off at 6.7 rad/sec. The filter in transfer function form is

$$G_{w_{SISO}}(s) = \frac{w}{\xi} = \frac{0.0187}{s+6.7} \tag{4.14}$$

The sensor noise filter was designed to closely match the assumed error in measuring a 1 G response. This assumed error was ± 0.04Gs or 4 percent. The filter is high pass with scaling to adjust the rms to 0.04. The filter in transfer function form is

$$G_{n_{SISO}}(s) = \frac{n}{\eta} = \frac{0.04(s+0.01)}{10(s+10)} \tag{4.15}$$

4.2.6 LQG/LTR Design. A brief discussion of Luke's [Luk93] LQG/LTR design is given here. This background is presented so that the design process for LQG/LTR is understood when comparing the mixed H_2/H_∞ controller to the LQG/LTR controller. For details on this design see [Luk93].

The first step in the design process is to develop a desired loop shape. A loop shape was chosen that had a maximum bandwidth of 10 radians/sec and a 20 dB/decade roll-off at bandwidth. In order to make the loop shape meet the specified barriers, a bank of integrators is added to the design plant. These integrators increase the order of the controller by two.

The next step was to design the Kalman filter and the regulator. The design parameters H, ρ, Λ and μ (which are the state weighting, the control weighting, the process noise distribution matrix, and the sensor noise intensity, respectively) were chosen by Luke [Luk93].The design variable H was set equal to the C matrix of the plant, ρ=1, Λ was set equal to the second column in the A matrix, and μ was chosen to be 100. The variable that affects recovery, q, was varied, with $q = 1000$ chosen. The resulting controller will be compared with mixed H_2/H_∞ controller designs in Section 4.4.

4.3 Mixed H_2/H_∞ Controller Design

The mixed H_2/H_∞ controller design method allows the designer to choose levels of tracking and noise rejection performance. The objective for this example is to calculate mixed H_2/H_∞ curves for different orders of the controller, and then compare their performance. The mixed controller designs at different orders will also be compared to the LQG/LTR design from the previous section. The mixed design is developed using the updated numerical optimization code discussed in Chapter 3. This section is broken into five parts. First, the H_2 and H_∞ problems are developed. Then different controller orders are analyzed for the mixed H_2/H_∞ design. The first controller order evaluated is the order of the H_2 plant. Next the mixed H_2/H_∞ controller with order of the H_∞ plant is examined, and finally, a reduced order controller one order less than the H_2 problem is analyzed.

4.3.1 H_2 Problem. The minimum order of the mixed H_2/H_∞ controller for which strong statements can be made about its characteristics is driven by the order of the H_2 plant. Therefore, the designer may wish to keep the order of the H_2 problem as low as possible. To keep the order as low as possible, it is best to use static weights on z and w. When static weights are used, the H_2 problem then becomes the standard LQG problem. This means the filters used for the noise will attenuate all frequencies equally. The values chosen for the static gains were Q_o=0.0028 for the process noise (i.e. wind) and R_f=0.004 for the sensor noise. The control weighting chosen was R_c=100 and H was chosen equal to the C matrix of the plant. The vector Γ corresponds to the column in the plant A matrix that multiplies the α state. The value Λ is equal to 1.

The H_2 problem for this example can then be written as

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_a \\ \dot{x}_d \end{bmatrix} = \overbrace{\begin{bmatrix} A_p & B_pC_a & 0 \\ 0 & A_a & 0 \\ B_dC_p & B_dD_pC_a & A_d \end{bmatrix}}^{A_2} \begin{bmatrix} x_p \\ x_a \\ x_d \end{bmatrix} + \overbrace{\begin{bmatrix} \Gamma\sqrt{Q_o} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}^{B_w} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \overbrace{\begin{bmatrix} B_dD_a \\ B_a \\ B_dD_pD_a \end{bmatrix}}^{B_{u_2}} u$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \overbrace{\begin{bmatrix} - & H & - \\ 0 & 0 & 0 \end{bmatrix}}^{C_z} \begin{bmatrix} x_p \\ x_a \\ x_d \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}^{D_{zw}} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ \sqrt{R_c} \end{bmatrix}}^{D_{zu}} u$$

$$y = \overbrace{\begin{bmatrix} D_dC_p & D_dD_pC_a & C_d \end{bmatrix}}^{C_{y2}} \begin{bmatrix} x_p \\ x_a \\ x_d \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & \Lambda\sqrt{R_f} \end{bmatrix}}^{D_{yw}} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \overbrace{\begin{bmatrix} D_dD_pD_a \end{bmatrix}}^{D_{yu}} u$$

The full state space matrices are in Appendix A.

4.3.2 H_∞ Problem. The tracking performance of the H_∞ controller for a step input is based on the weighting put on the system sensitivity. Usually this weighting is chosen to be the inverse of the desired sensitivity. Since a bandwidth of close to 10 rad/sec was required to compare to the LQG/LTR design, the weighting chosen was

$$W_s(s) = \frac{s + 8.0001}{s + 0.0001} \tag{4.16}$$

Figure 4.3 shows that the loop shape meets the boundaries chosen for the target loop shape. In state space the weight can be written as

$$A_s = [-0.0001] \quad B_s = [1] \quad C_s = [8] \quad D_s = [1]$$

Now, with the sensitivity weight added to the H_∞ system, the resulting complete H_∞ state space matrices can be written as

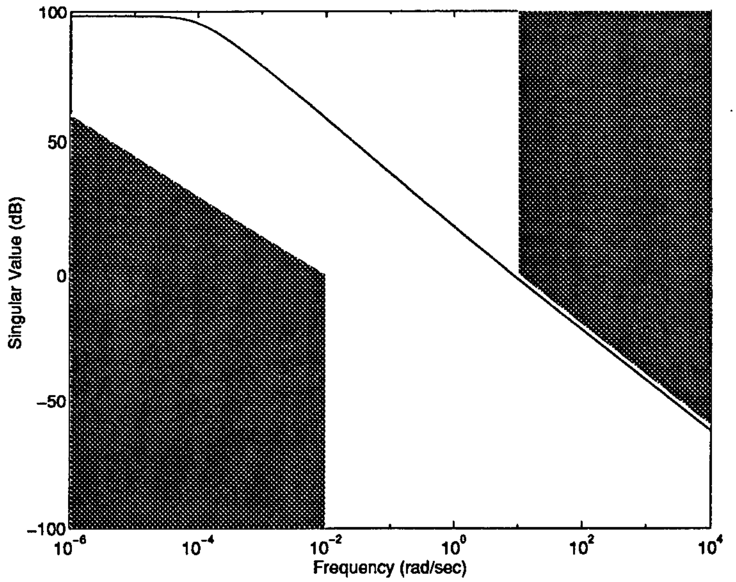

Figure 4.3 Target Loop Shape for H_∞ Problem

$$\begin{bmatrix}\dot{x}_p\\ \dot{x}_a\\ \dot{x}_d\\ \dot{x}_s\end{bmatrix} = \overbrace{\begin{bmatrix} A_p & B_pC_a & 0 & 0\\ 0 & A_a & 0 & 0\\ B_dC_p & B_dD_pC_a & A_d & 0\\ B_sD_dC_p & B_sD_dD_pC_a & B_sC_d & A_s\end{bmatrix}}^{A_\infty} \begin{bmatrix}x_p\\ x_a\\ x_d\\ x_s\end{bmatrix} + \overbrace{\begin{bmatrix}0\\ 0\\ 0\\ B_s\end{bmatrix}}^{B_d} d + \overbrace{\begin{bmatrix}B_dD_a\\ B_a\\ B_dD_pD_a\\ B_sD_dD_pD_a\end{bmatrix}}^{B_{u\infty}} u$$

$$e = \overbrace{\begin{bmatrix} D_sD_dC_p & D_sD_dD_pC_a & D_sC_d & C_s\end{bmatrix}}^{C_e} \begin{bmatrix}x_p\\ x_a\\ x_d\\ x_s\end{bmatrix} + \overbrace{\begin{bmatrix} D_s \end{bmatrix}}^{D_{ed}} d + \overbrace{\begin{bmatrix} D_sD_dD_pD_a \end{bmatrix}}^{D_{eu}} u$$

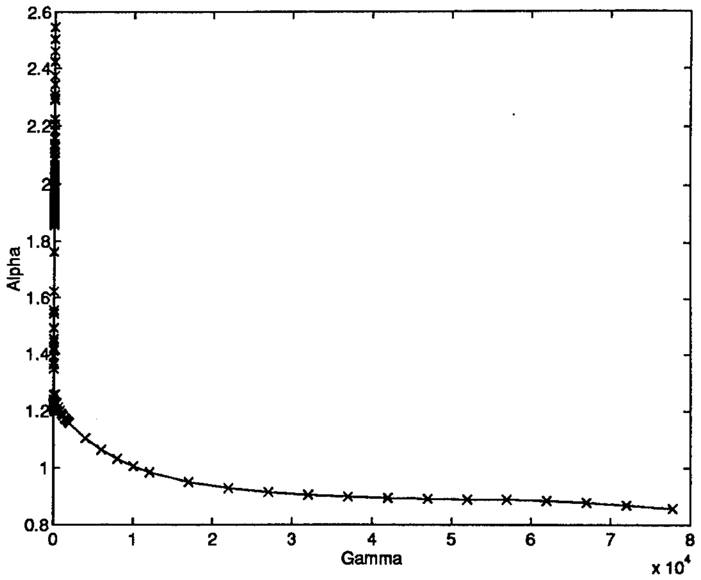

Figure 4.4 Mixed H_2/H_∞ Curve (Order H_2)

$$y = \overbrace{\begin{bmatrix} D_dC_p & D_dD_pC_a & C_d & 0 \end{bmatrix}}^{C_{y\infty}} \begin{bmatrix} x_p \\ x_a \\ x_d \\ x_s \end{bmatrix} + \overbrace{\begin{bmatrix} 1 \end{bmatrix}}^{D_{yd}} d + \overbrace{\begin{bmatrix} D_dD_pD_a \end{bmatrix}}^{D_{yu}} u$$

The full state space matrices for this example are in Appendix A. Since the sensitivity weight added to this problem is not strictly proper, the D_{ed} term is not zero. Also, D_{eu}=0, which means the problem is singular. However, from Walker's development [Wal94] there is no requirement for D_{ed} to be zero and singular H_∞ problems are allowed.

4.3.3 H_2 Order Mixed H_2/H_∞ Solution. The optimization routine was initiated at the H_2 optimal design. The calculation of the H_2 order mixed controller previously required days to compute only six points on the mixed H_2/H_∞ curve. The new code generated over 150 controllers in less than 3 hours. A plot of the mixed curve is shown in Figure 4.4.

Note that the lowest value of γ achieved for this example was 1.361. The minimum value of γ found for the H_∞ problem, which is one order higher than the H_2 order, was

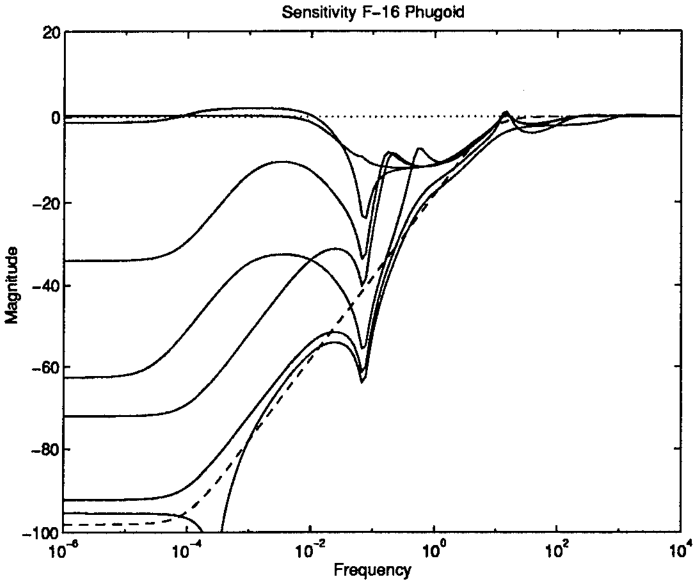

Figure 4.5 Sensitivity of Closed Loop System : Dashed Line - Target Loop Shape (Order H_2)

$\underline{\gamma}$=1.3201. This value should not be expected to be achieved, since this is for a controller of the H_∞ order. As the order of the controller is decreased, $\underline{\gamma}$ increases.

Sensitivity curves for this problem are plotted in Figure 4.5. This shows the trade-off between H_2 and H_∞. As γ is decreased, the H_∞ part of the problem becomes more dominant and the sensitivity plot approaches the target loop shape, (i.e., the dashed line on Figure 4.5). The margins for the problem can be found using this plot. Notice the peak is relatively low indicating good margins.

4.3.4 H_∞ Order Mixed H_2/H_∞ Solution. Since the optimal value of $\underline{\gamma}$ at the H_∞ order of the plant is known to be 1.3201, the optimizer at this order should get very close to this value. It was desired to look at the effect of performance of a mixed controller as close as possible to this optimal $\underline{\gamma}$ level. The optimization code must be started with a controller of the appropriate order. The previously generated controllers have order one less than required here. The order of the controller can be increased to provide an acceptable starting point by adding a near pole-zero cancellation or by adding a pole beyond the bandwidth of the system. The second approach was chosen. A lag was added to a controller found for the H_2 order

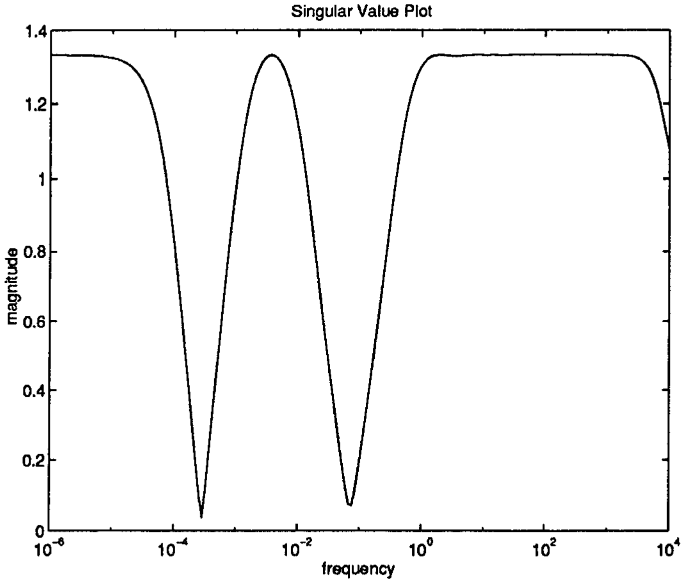

Figure 4.6 Singular Value Plot During Optimization (Order H_∞)

problem with a γ=1.5. Consequently, the lag increased γ for the starting controller from γ=1.5 to γ=1.58. This lag was

$$\frac{1000}{s+1000} \tag{4.17}$$

Running the optimization code at the order of the H_∞ problem should allow the optimizer to get very close to the optimal $\underline{\gamma}$ level. The final γ level achieved for this problem was 1.331. Figure 4.6 shows the resulting singular value plot of the closed loop system for γ=1.331. There were six peaks constrained for that singular value plot. It should be noted how flat the singular value plot has become. A plot of the mixed H_2/H_∞ curve at the H_∞ order for the controller is shown in Figure 4.7.

The sensitivity plot for this order controller shows that the low frequency dip is removed as γ is decreased. It also shows the higher frequency dip in the sensitivity is not removed. This plot is given in Figure 4.8.

4.3.5 Reduced Order H_2/H_∞ Solution. The final question is what is the controller performance when the order of the controller is reduced from H_2 optimal. A direct technique

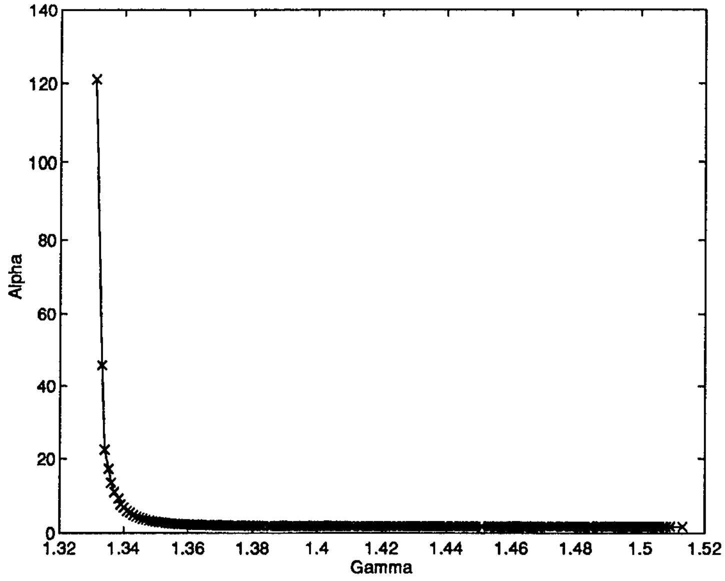

Figure 4.7 Mixed H_2/H_∞ Curve (Order H_∞)

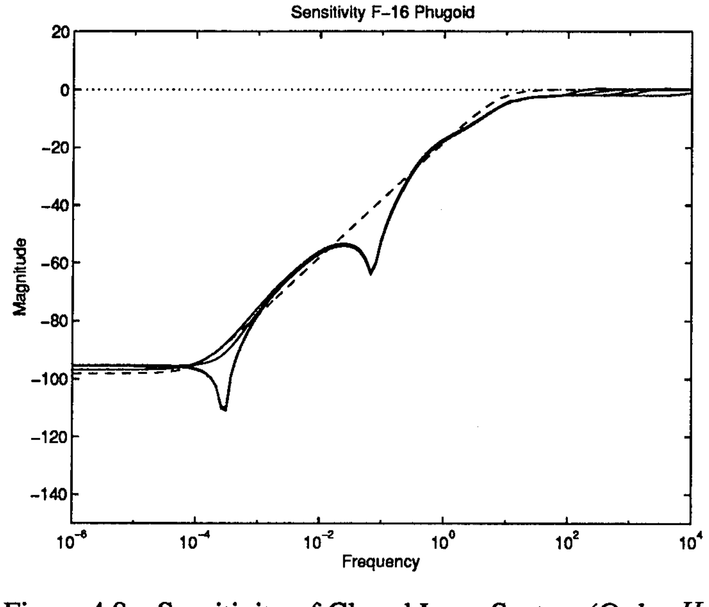

Figure 4.8 Sensitivity of Closed Loop System (Order H_∞)

was used to reduce the order of the controller. This was accomplished by running the MATLAB function 'SCHMR'. This function uses the Schur method for model reduction. The function SCHMR constrains the infinity-norm of the difference between the original and reduced transfer functions to be less than a specified value. The designer also has the option of choosing the order of the reduced transfer function. There is no guarantee that the controller is

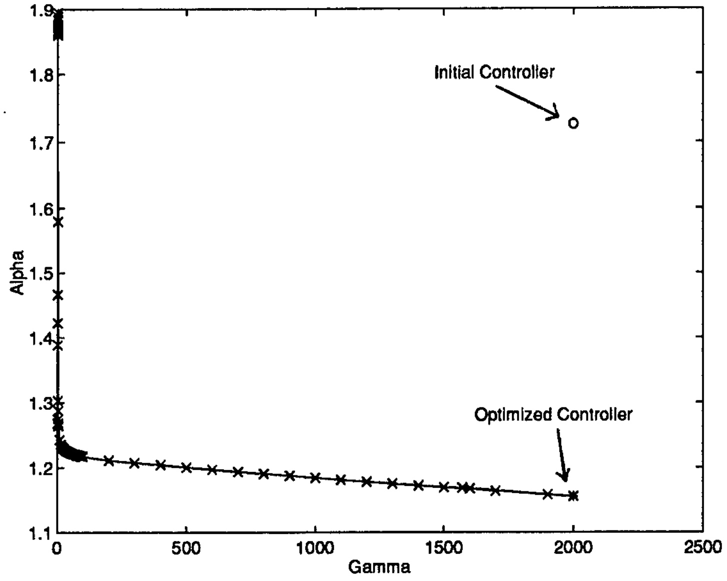

Figure 4.9 Mixed H_2/H_∞ Curve (Order Reduced)

stabilizing after this model reduction routine has been performed. The reduced order controller was formed from a mixed H_2/H_∞ design that was the order of the H_2 problem. This controller had a γ=2000 and α=1.156. This controller was found to be both stable and stabilizing for the closed loop system. The mixed H_2/H_∞ curve for the reduced order problem is shown in Figure 4.9. The point on the graph marked with an 'o' was the starting point for the reduced order controller. At this point, the two-norm was 1.725 and the infinity-norm was 2000. Note the infinity-norm did not change from the sixth order controller to the fifth order. The optimum two-norm value was found for this controller by setting the γ step level equal to zero and letting the optimizer find the controller with the lowest two-norm. The optimum two-norm for the starting controller is marked by a '*' on Figure 4.9 at a value of 1.156.

The H_2 optimal controller at orders less than the plant is no longer unique. There may be many different controllers at this order that give the same H_2 norm, but may have different H_∞ norms for T_{ed}. No attempt was made to find an optimal reduced order H_2 controller here; rather, the starting point described above was used.

A plot of the closed loop sensitivity is shown in Figure 4.10. The sensitivity plot for reduced order shows a slight improvement from the H_2 order in being able to match the target

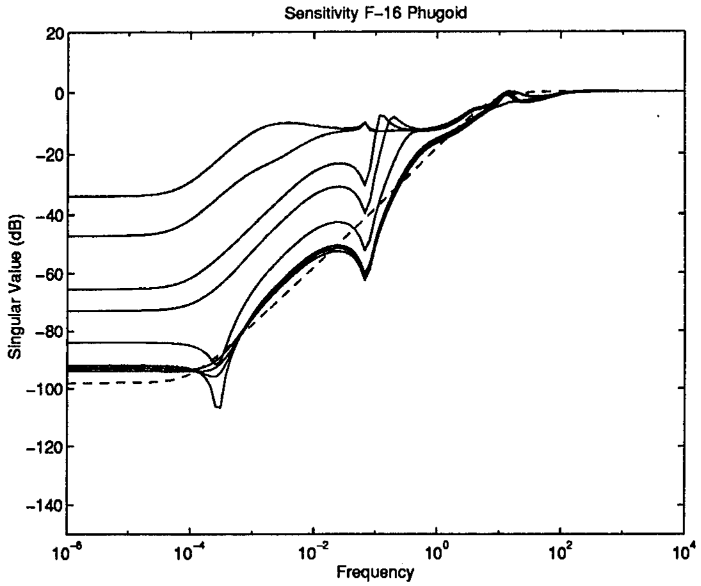

Figure 4.10 Sensitivity of Closed Loop System (Order Reduced)

loop shape. The phugoid dip in the sensitivity plot is removed completely at the lowest γ controller for the reduced design. The H_2 order sensitivity plot shows the short period dip still remains at the lowest γ level achieved.

4.4 Comparison of Controller Designs

Each of the mixed H_2/H_∞ designs was compared to the LQG/LTR design discussed in Section 4.2.6. Simulations were run on MATLAB SIMULINK. Comparisons were made for each controller's closed-loop response to a unit step and an initial 5 degree perturbation in angle of attack.

4.4.1 Mixed H_2 Order vs LQG/LTR . This section presents the comparison of the LQG/LTR controller versus the H_2 order mixed H_2/H_∞ controller. Figure 4.11 shows the mixed curve for the H_2 order controller. The LQG controller is shown with the symbol 'o' in Figure 4.11. Note that the LQG two-norm is higher than that of H_2; this is due to the augmented integrators in the LQG design. Since Loop Transfer Recover techniques were used to improve the tracking of the LQG design, the LQG/LTR design is shown on the mixed

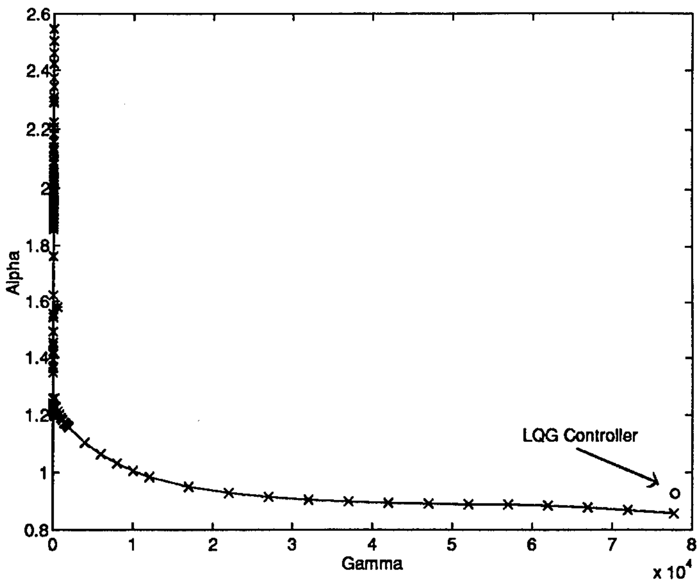

Figure 4.11 Mix H_2/H_∞ Curve with LQG design (Order H_2)

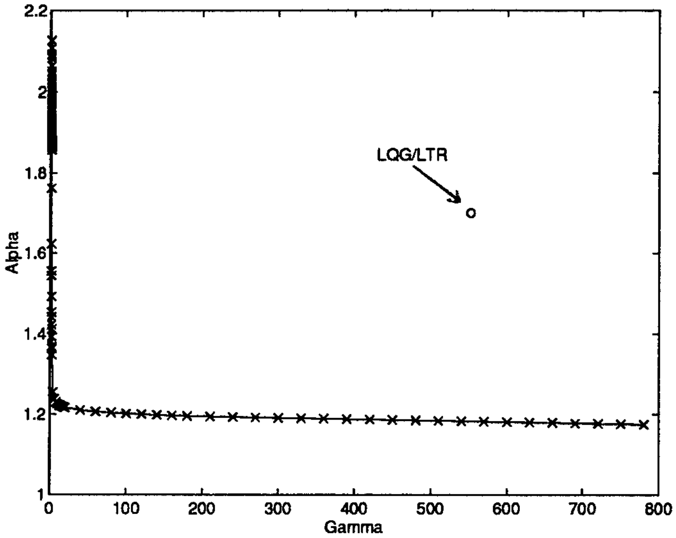

Figure 4.12 Mix H_2/H_∞ Curve with LQG/LTR design (Order H_2)

H_2/H_∞ curve in Figure 4.12. It is important to note that the LQG and LQG/LTR designs are eighth order where the mixed design shown here is sixth order.

Figure 4.12 shows that the eighth order LQG/LTR controller can be improved in both noise performance and tracking compared to the mixed 6th order controller. The LQG/LTR design resulted in α=1.58 and γ=551.88. A mixed controller was chosen which resulted in

Table 4.1 Gain and Phase Margin Comparison (H_2 Order)

Mixed H_2/H_∞					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.4 9.8]	± 39.5	[-10.3 4.5]	± 40.8	[-10.3 9.8]	± 40.8

LQG/LTR					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.6 10.6]	± 41.35	[-6.5 3.7]	± 30.8	[-6.5 10.6]	± 41.35

α=1.76 and γ=1.5 to compare to the LQG/LTR design. The gain and phase margins were calculated for the mixed H_2/H_∞ controller and the LQG/LTR controller, and are shown in Table 4.1. The total phase margins are virtually the same for both controllers, but the mixed H_2/H_∞ total gain margins are slightly better than LQG/LTR.

Figure 4.13 shows the response of both controllers for an initial 5 degree perturbation in angle of attack and a 1 G unit step in N_z. The noise characteristics look about the same but the mixed H_2/H_∞ controller is a better tracker. The mixed controller takes less time to settle for the response to an initial five degree angle of attack. It also shows less of an overshoot for the step-response.

4.4.2 Mixed H_∞ Order vs LQG/LTR . The mixed H_2/H_∞ curve at the order of the H_∞ problem is shown in Figure 4.14. The point 'o' on the curve is the LQG/LTR 8th order design. As in the previous example, there is a mixed design at 7th order that improves the two-norm and infinity-norm of the system compared to the LQG/LTR controller. The mixed controller chosen for comparison produced a two-norm equal to 1.66 and an infinity-norm equal to 1.5. The gain and phase margins were calculated for the mixed H_2/H_∞ controller. These are in Table 4.2. The margins for H_∞ order are approximately the same as the margins for H_2 order.

The response of the mixed design at H_∞ order is shown in Figure 4.15. The response of the mixed design demonstrates better performance than the LQG/LTR design. The mixed

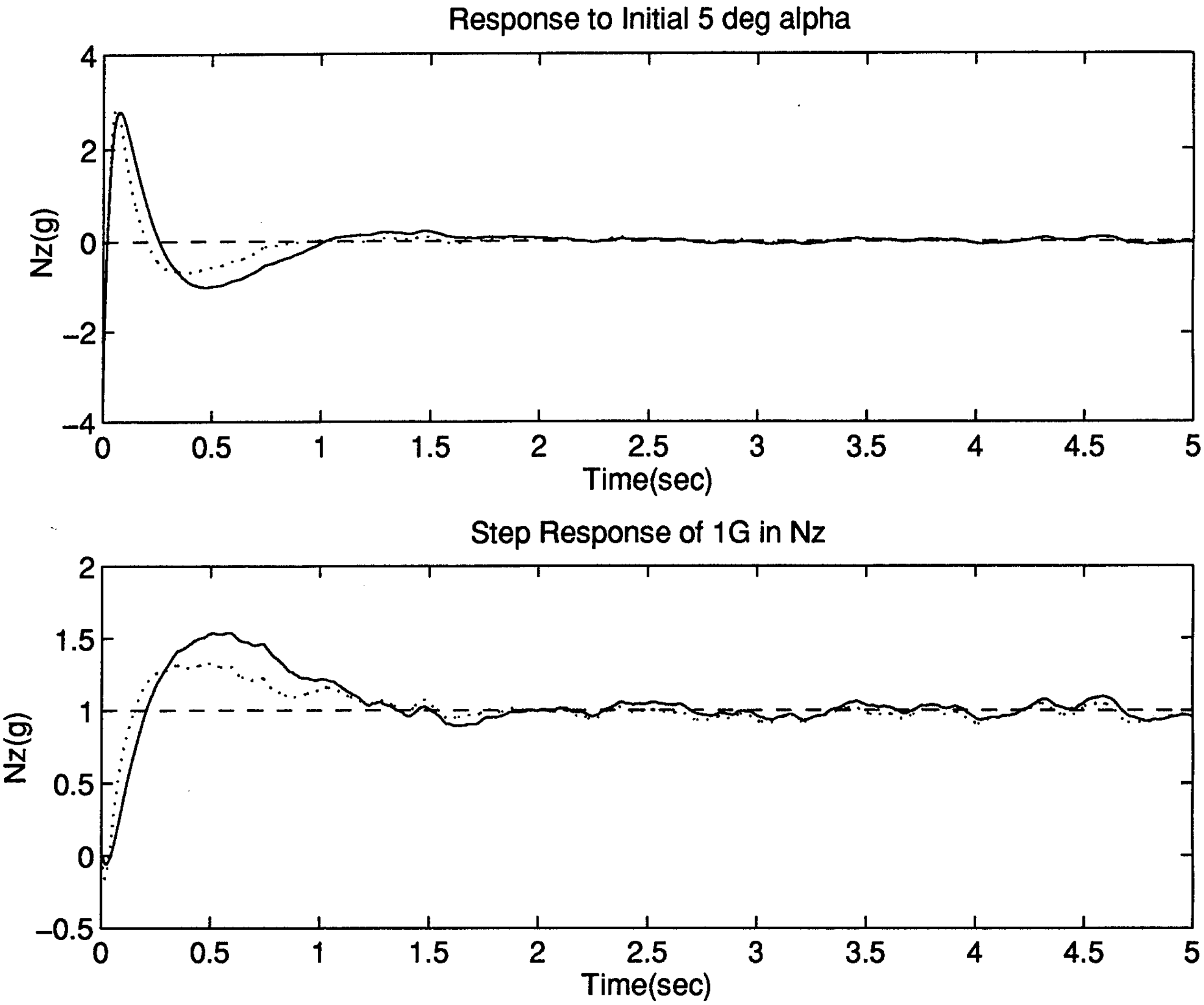

Figure 4.13 Responses of Mixed H_2/H_∞ (Order H_2) (dash) vs LQG/LTR (solid)

Table 4.2 Gain and Phase Margin Comparison (H_∞ Order)

Mixed H_2/H_∞					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.4 9.6]	± 39.2	[-10.3 4.5]	± 40.7	[-10.3 9.6]	± 40.7

LQG/LTR					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.6 10.6]	± 41.35	[-6.5 3.7]	± 30.8	[-6.5 10.6]	41.35

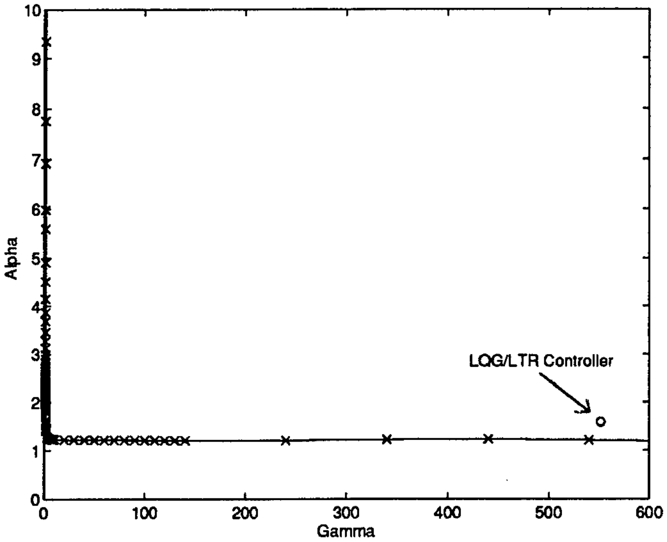

Figure 4.14 Mix H_2/H_∞ Curve with LQG/LTR design (Order H_∞)

controller, as in the H_2 order, has a quicker settling time for the initial condition and less of an overshoot for the step response. The noise performance is approximately the same. Overall, there is no real benefit in using the H_∞ order versus the H_2 order mixed controller.

4.4.3 Mixed Reduced Order vs LQG/LTR . The mixed H_2/H_∞ curve for one order lower than the H_2 problem (order 5) is shown in Figure 4.16. The mixed controller chosen for comparison gave a two-norm of 1.89 and an infinity-norm of 1.59. The LQG/LTR design is also plotted on this curve for comparison purposes. It can be seen that even at this lower order for the mixed controller the two-norm and infinity-norm can be improved over the LQG/LTR design. The gain and phase margins were also calculated for this mixed reduced order H_2/H_∞ controller. They are in Table 4.3. Both the gain and phase margins are slightly better for the LQG/LTR design than the mixed H_2/H_∞ design. The lower order mixed margins are slightly worse than the H_2 and H_∞ order margins with little degredation in tracking performance. This was expected at reduced order.

The response performance for the reduced order controller versus the LQG/LTR design is shown in Figure 4.17. This shows that the mixed design has a faster response time to an initial input of a 5 degree alpha command and the unit step response to a 1 G input of Nz has

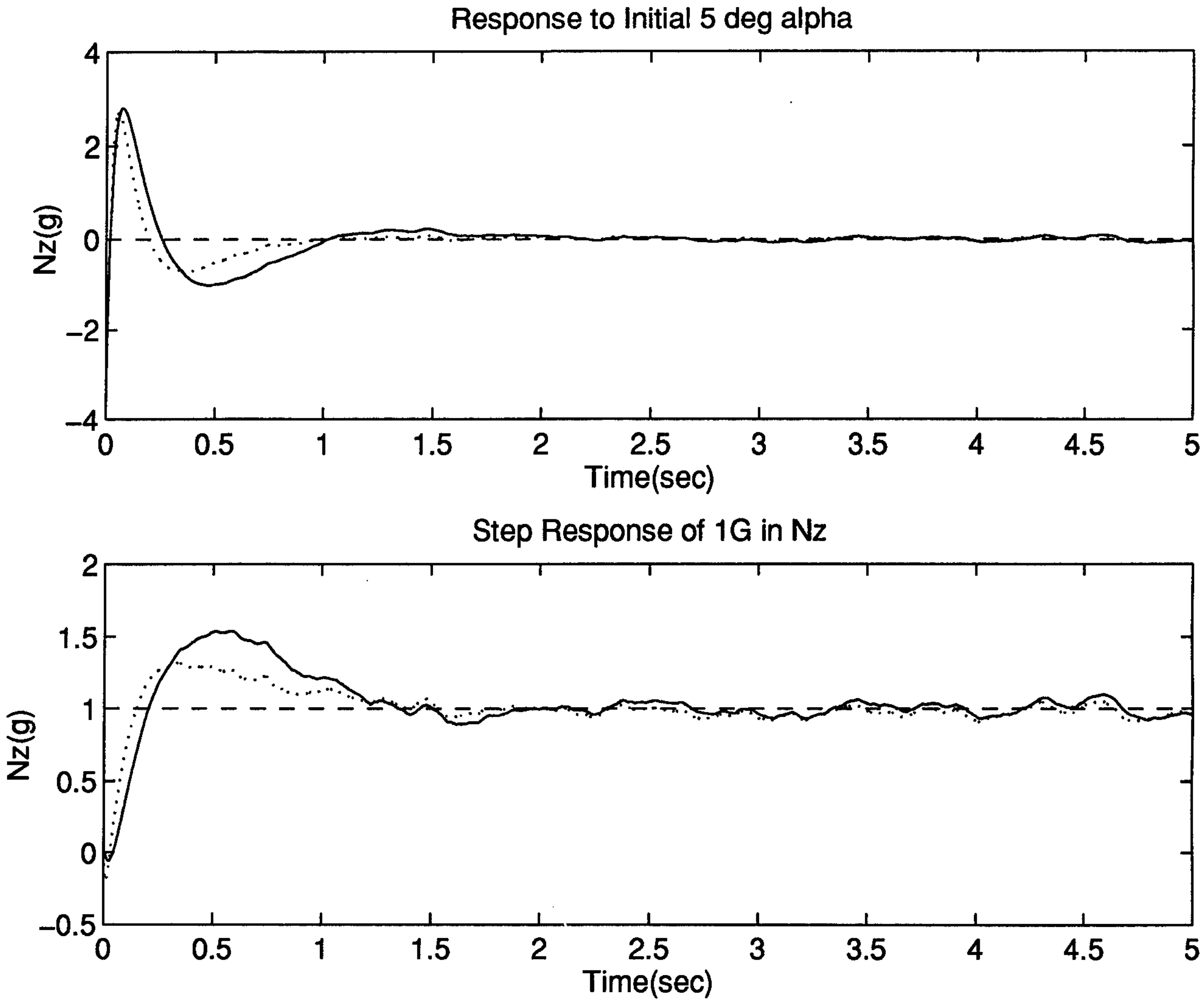

Figure 4.15 Responses of Mixed H_2/H_∞ (Order H_∞) (dash) vs LQG/LTR (solid)

Table 4.3 Gain and Phase Margin Comparison (5th Order)

Mixed H_2/H_∞					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.3 9.23]	± 38.2	[-9.9 4.5]	± 39.8	[-9.9 4.5]	± 39.8

LQG/LTR					
VGMs (dB)	VPMs (deg)	VGMt (dB)	VPMt (deg)	Total VGM (dB)	Total VPM (deg)
[-4.6 10.6]	± 41.35	[-6.5 3.7]	± 30.8	[-6.5 10.6]	± 41.35

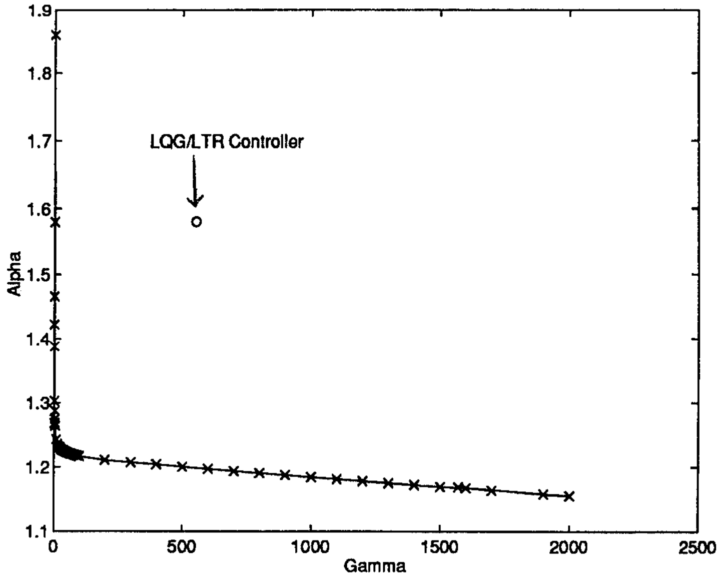

Figure 4.16 Mix H_2/H_∞ Curve with LQG design (Reduced Order)

less overshoot. This mixed H_2/H_∞ design of order five performs better than the LQG/LTR design of order eight.

4.4.4 Summary of Comparisons. The purpose of this example was to determine the effectiveness of mixed H_2/H_∞ optimization versus LQG/LTR for a real-world example. The results from this example show the mixed approach is better than LQG/LTR design method. This example demonstrated that the mixed H_2/H_∞ controller at three orders less than the LQG/LTR design is a better tracker. The noise response for the different order mixed controllers versus the LQG/LTR remained relatively the same. The optimal infinity-norm at seventh order was 1.33, sixth order was 1.36 and fifth order was 1.59. This similar noise and tracking response between the different order controllers is due to choosing mixed controllers with the same approximate two-norm and infinity-norm. Although this was not done by design, the infinity-norm for the mixed controllers was 1.5, 1.5 and 1.59. The two-norm for the designs chosen were 1.66, 1.76 and 1.89 for seventh, sixth and fifth order mixed H_2/H_∞ controllers. Each choice was based on achieving the best two-norm and infinity-norm for that particular curve.

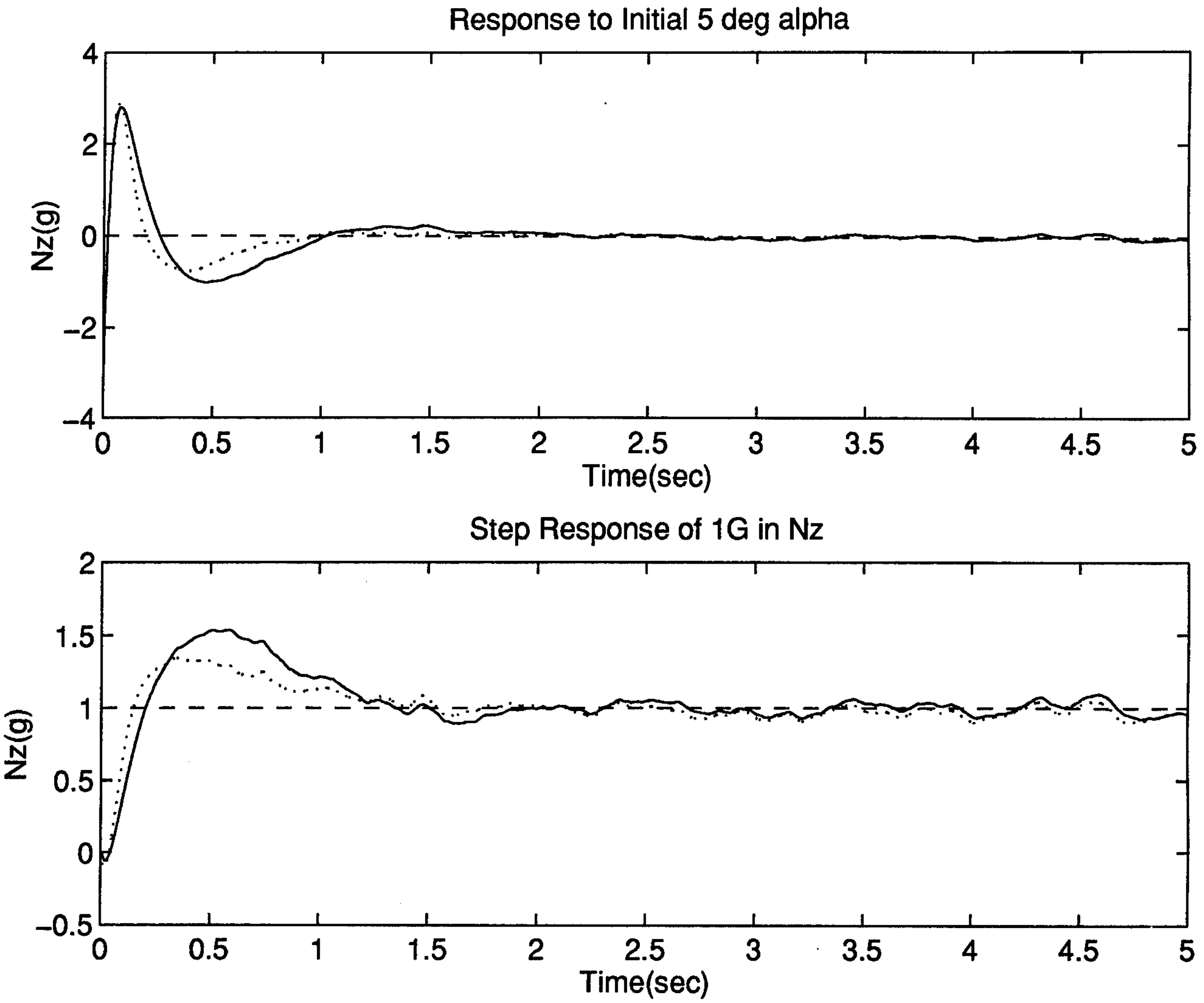

Figure 4.17 Responses of Mixed H_2/H_∞ (Reduced Order) (dash) vs LQG/LTR (solid)

The results presented in this chapter also demonstrate the improvement in the H_2/H_∞ optimization code. This optimization improvement provides the capability to trade off H_2 and H_∞ objectives for the entire mixed H_2/H_∞ curve. Previously the mixed solution could not generate enough of the curve in the region of the H_∞ optimal mixed solution to examine the full tradeoff.

The results of this example showed improvement in performance at lower order controllers. The results also demonstrated the improved numeric capability to generate the full mixed H_2/H_∞ curve in significantly less time than was previously required. These were important for a SISO example; however, these improvements become much more significant for a MIMO example. The missile MIMO problem is addressed in Chapter V.

V. MIMO Example

The control problem discussed in this chapter is for an air-to-air missile that uses tail-control fins. The purpose of this chapter is to design a mixed H_2/H_∞ controller for this example and then compare that design to a LQG/LTR design previously completed by Brown [Bro91]. The mixed H_2/H_∞ design was attempted with the previous optimization code, but due to numerical difficulties no results could be generated. The improved optimization code was applied to this MIMO problem to test its effectiveness. The full H_2/H_∞ curves could be generated for this problem. Since the goal is to make a comparison between the LQG/LTR and the mixed controller designs, this example concentrates on one point in the missile flight envelope.

5.1 Background

The added challenge of the controller design for this example is that the plant is non-minimum phase with lightly damped complex poles. Although this is usually a difficult problem for LQG/LTR, Brown successfully calculated the LQG/LTR controller using the method of eigenstructure reassignment through static feedback. This static feedback matrix does not add to the dimension of the plant, therefore keeping the controller order at a minimum. Eigenstructure reassignment was used first on the plant to shift the poles and then LQG/LTR was applied to the problem. This method proved beneficial to the design.

5.1.1 Eigenstructure reassignment. It has been shown by Andry [ASC83] that complete pole reassignment is possible if the system is completely observable, completely controllable, and $n \leq r + m - 1$ where n is the order of the system, r is the number of inputs, and m is the number of outputs. This missile MIMO example did not require all the poles to be moved. A method developed in [Bro91] allows a limited number of the system's poles to be reassigned with the remaining poles drifting to unassigned positions. This method also allows arbitrary assignment of portions of the system's eigenvectors.

The method begins with the state space representation of a completely controllable completely observable system.

$$\begin{aligned}\dot{\mathsf{x}} &= \mathcal{A}\mathsf{x} + \mathcal{B}\mathsf{u} \\ \mathsf{y} &= \mathcal{C}\mathsf{x}\end{aligned} \tag{5.1}$$

where A is of rank n, B is rank m, and C is rank r. Note this development assumes that there is no D matrix.

The problem then is to calculate a constant gain matrix K_{stat} that has dimensions mxr, which will reassign the poles through inner loop feedback. The state space equation with K_{stat} is now

$$\begin{aligned}\dot{\mathsf{x}} &= [\mathcal{A} + \mathcal{B}\mathsf{K}_{stat}\mathcal{C}]\,\mathsf{x} + \mathcal{B}\mathsf{u} \\ \mathsf{y} &= \mathcal{C}\mathsf{x}\end{aligned} \tag{5.2}$$

5.1.2 Eigenvector assignability. The definition of eigenvector assignability was addressed by Andry [ASC83]. He discusses the extent of possible eigenvector specification for the closed-loop system. The definition of the eigenvalue, eigenvector pair (λ_i and v_i) is given as:

$$[\mathcal{A} + \mathcal{B}\mathsf{K}_{stat}\mathcal{C}]\,\mathsf{v}_i = \lambda_i \mathsf{v}_i \tag{5.3}$$

or, rearranged,

$$\mathsf{v}_i = [\lambda_i I - \mathcal{A}]^{-1}\,\mathcal{B}\mathsf{K}_{stat}\mathcal{C}\mathsf{v}_i \tag{5.4}$$

It is required, from the development, that the eivenvector v_i must lie in the subspace spanned by the columns of $[\lambda_i I - \mathcal{A}]^{-1}\mathcal{B}$ which is rank B. Since the desired eigenvector will probably not be in the exact subspace defined an achievable eigenvector is calculated by a projection of the desired vector onto the subspace spanned by the columns of $[\lambda_i I - \mathcal{A}]\mathcal{B}$.

Only some of the components of the eigenvector are reassignable. The desired assignable components are chosen by the designer to try to affect performance goals. The desired eigenvector can be reordered to contain the assignable components at the top of the vector and the unassignable components at the bottom of the vector. This allows the calculation of the assignable eigenvector.

5.1.3 K_{stat} Calculation. The gain matrix K_{stat} is developed assuming the eigenvectors are in the assignable general subspace. The process of forming K_{stat} begins by transforming the system so that the B matrix becomes:

$$\mathcal{B} \rightarrow \mathcal{B}' = \begin{bmatrix} [I_m] \\ \cdots \\ [0] \end{bmatrix} \tag{5.5}$$

A similarity transformation matrix, T, is formed such that

$$\mathsf{x} = T\mathsf{x}' \tag{5.6}$$

where

$$T = \begin{bmatrix} [\mathcal{B}] & \vdots & [P] \end{bmatrix} \tag{5.7}$$

This similarity transformation allows the eigenvalues of the system to be unchanged but the eigenvectors have been transformed by:

$$\mathsf{v}_i' = T^{-1}\mathsf{v} \tag{5.8}$$

The equation for the closed-loop eigenvalues and eigenvectors written in partitioned form becomes

$$\begin{bmatrix} [\lambda_i I_m - \mathcal{A}_{11}] & [-\mathcal{A}_{12}] \\ [-\mathcal{A}_{21}] & [\lambda_i I_{n-m} - \mathcal{A}_{22}] \end{bmatrix} \begin{bmatrix} [\eta_i] \\ [\mathsf{w}_i] \end{bmatrix} = \begin{bmatrix} [I_m] \\ [0] \end{bmatrix} \mathsf{K}_{stat}\mathcal{C} \begin{bmatrix} [\eta_i] \\ [\mathsf{w}_i] \end{bmatrix} \tag{5.9}$$

Using the top equation, rearranging and defining $\mathcal{A}_1 = \left[\mathcal{A}_{11} \vdots \mathcal{A}_{12}\right]$ the equation can be written as

$$[\mathcal{A}_1 + \mathsf{K}_{stat}\mathcal{C}]\, \mathsf{v}_i = \lambda_i \eta_i \tag{5.10}$$

This equation must be true for all eigenvalue/assignable eigenvector pairs. Rearranging (5.10) and solving for K_{stat} gives the equation

$$\mathsf{K}_{stat} = [\mathsf{N} - \mathcal{A}_1 \mathsf{W}]\, [\mathcal{C}\mathsf{W}]^{-1} \tag{5.11}$$

5.2 Missile Model

The missile is assumed rigid and the missile's airframe dynamics are modeled using six state variables; u, α, β, p, q, and r. These variables represent forward velocity in feet/second, angle of attack in the missile's pitch-plane in radians, sideslip angle in radians, roll rate, pitch rate, and yaw rate in radians/second. The model is driven by four fin inputs: δ_1, δ_2, δ_3, and δ_4. The outputs are p, q, r, N_x, N_y, and N_z. The outputs p, q and r are measurements from the rate gyros of roll, pitch and yaw respectively. The terms N_x, N_y and N_z are accelerometer measurements from the rate gyros in the x, y and z directions repsectively.

Brown developed the linearized equations of motion for this problem. The linearized equations were compared to the sponsor's six degree of freedom (6-DOF) model. Overall, it was determined that the linearized model was adequate for the LQG/LTR study. These equations of motion were put into standard state space format using the notation

$$\begin{aligned} \dot{\mathsf{x}} &= \mathsf{Ax} + \mathsf{Bu} \\ \mathsf{y} &= \mathsf{Cx} + \mathsf{Du} \end{aligned} \tag{5.12}$$

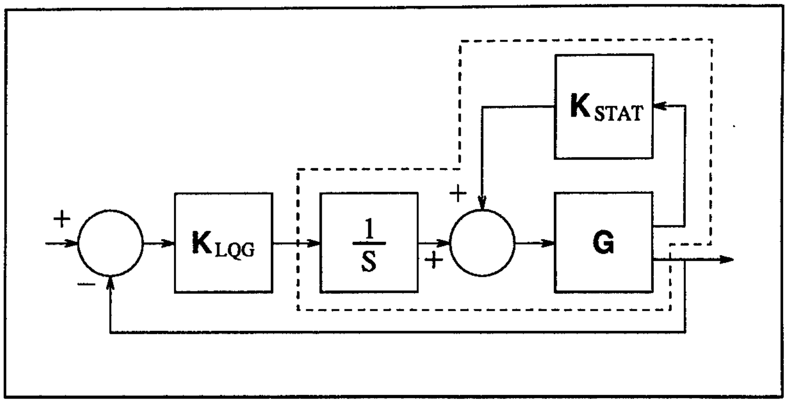

Figure 5.1   Schematic of Closed-Loop System for Missile

where

$$\mathbf{x} \equiv \begin{bmatrix} u \\ \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}, \quad \mathbf{u} \equiv \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix}, \quad \mathbf{y} \equiv \begin{bmatrix} p \\ q \\ r \\ N_x \\ N_y \\ N_z \end{bmatrix}$$

The A, B and C matrices for this problem are shown in Appendix B.

5.3 LQG/LTR Design

The design process includes evaluating performance boundaries, evaluating the need for augmenting the plant with integrators, eigenstructure reassignment and the filter and regulator design. The closed-loop system can be represented by Figure 5.1. The inputs to K_{stat} are p, q, r, N_y and N_z. The output of K_{stat} is then fed directly into the plant before the bank of integrators. The performance bounds for this design include a 68 percent rise time in normal acceleration within 200ms. Another requirement is minimum overshoot. A final requirement is zero steady state error for a step input.

The system response equation was then defined by Brown. The particular quantities that were to be controlled were roll rate(p), y-axis acceleration (N_y) and z-axis accleration (N_z). This is a subset of the system outputs. Defining the response equation in this manner effectively discards the output information of the two states. The system response equation is:

$$\mathsf{z} = \mathsf{H}\mathsf{x} = \begin{bmatrix} p \\ N_y \\ N_z \end{bmatrix} \tag{5.13}$$

Order reduction of the problem was first accomplished by recognizing that the first state, u was totally decoupled from the other states and did not appear in any of the equations for the system response inputs. Therefore, the linear model could be reduced by one state. This was done by eliminating the rows in the state space A, B, and C matrices that corresponded with the state u. Finally an order reduction in the plant was achieved by reducing the order of the model of the actuator dynamics. The actuator dynamics of the missile can be modeled as

$$\begin{aligned} \dot{\mathsf{x}}_m &= \mathcal{A}_m \mathsf{x}_m + \mathcal{B}_m \mathsf{u}_m \\ \mathsf{y}_m &= \mathcal{C} \mathsf{x}_m + \mathcal{D}_m \mathsf{u}_m \end{aligned} \tag{5.14}$$

where

$$\mathsf{u}_m = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \tag{5.15}$$

Brown reduced the order of the actuator models. The original model presented by the sponsor was second order with high damping ($\zeta = 0.8$) and undamped natural frequency of 408 rad/sec. He showed that a useful approximation of this actuator was a first order model with the pole at s=-400. The state space representation of the plant augmented with the actuators is given as:

$$\begin{bmatrix} \dot{\mathsf{x}}_m \\ \dot{\mathsf{x}}_a \end{bmatrix} = \begin{bmatrix} \mathcal{A}_m & \mathcal{B}_m\mathcal{C}_a \\ 0 & \mathcal{A}_a \end{bmatrix} \begin{bmatrix} \mathsf{x}_m \\ \mathsf{x}_a \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{B}_a \end{bmatrix} \mathsf{u}_a$$

$$\mathsf{y}_m = \begin{bmatrix} \mathcal{C}_m & \mathcal{D}_m\mathcal{C}_a \end{bmatrix} \begin{bmatrix} \mathsf{x}_m \\ \mathsf{x}_a \end{bmatrix} \tag{5.16}$$

The four fin deflections (δ_1 through δ_4) were expressed in standard fin deflections ($\delta_p, \delta_q, \delta_r$). These were the required fin deflections to produce roll-rate, pitch-rate, and yaw-rate respectively. The transformation from four inputs to three was done by mixing the fin deflections accordingly

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix} \tag{5.17}$$

5.3.1 Design for Condition 7. Flight condition 7 represents the missile flying at Mach 2 near sea level. Brown did this design both with and without eigenstructure reassignment. This discussion will only present the design with the eigenstructure reassignment since that design performed better.

The eigenvector reassignment was done with a goal of achieving specific relationships between the states. Reassigning the defined components for the eigenvectors is the same for each flight condition. The one real pole was moved to the left slightly to keep gains low. The two complex pairs of poles were moved slightly to the left, increasing the natural frequency and particularly the damping. Original poles (-1.6, $-1.22 \pm j13.9$ and $-1.31 \pm j14.0$). The new poles were (-2, $-15 \pm j15.0$ and $-15 \pm j15.0$).

The filter was designed with a value of μ=3000 to accomplish a 30 dB lowering of the singular value plot. The controller design used q^2=10000 to get the best loop shape. This design only slightly violated the left boundary for the loop shape.

5.4 Mixed H_2/H_∞ Design

The mixed H_2/H_∞ design was approached in two ways. The first formulation follows Brown's LQG/LTR design method where two of the five outputs are discarded. The two measurements not used were q and r. This will be referred to as the 3x3 MIMO problem. The second approach is not to ignore the other two measurements and include them in the mixed H_2/H_∞ design. This will be called the 5x3 MIMO problem.

The LQG/LTR design by Brown is 14th order, where the plant to be controlled is eighth order. Since the mixed H_2/H_∞ design is the same order of the plant of the H_2 problem, the resulting controller will be order eight.

5.4.1 3x3 MIMO Example. This development ignores two measurements. This problem is broken into the separate H_2 and H_∞ problems.

5.4.1.1 H_2 Problem. The goal of the H_2 problem is to keep the order of the problem as low as possible since that will be the order of the controller for the mixed design. The weights that will be used are static and again as in the SISO example the set-up becomes the standard LQG problem. The order for the H_2 problem is then eight.

The weight chosen for the LQG design was $H = C_{plant}$, so that $Q_c = H^T H$. The control weighting R_c was equally weighted for each fin. This was done due to the fact that the missile is symmetric about its longitudinal axis. Thus, $R_c = \rho I$ where ρ=5000 for this design.

The Kalman filter design used MIMO noise models. These models were developed in the same way as the SISO noise model development in Chapter 4. The difference between the wind noise model for the SISO problem and the noise model for the MIMO problem was that a perturbation in β must also be added. The process noise was modeled as

$$G_{w_{MIMO}} = \begin{bmatrix} 2\text{x}10^{-3} & 0 \\ 0 & 2\text{x}10^{-4} \end{bmatrix} \tag{5.18}$$

and no filter dynamics were added. The sensor noise was developed in exactly the same way as the SISO problem. The sensor noise model used was

$$G_{n_{MIMO}} = \frac{n}{\eta} = \begin{bmatrix} \frac{0.1}{(32)(57.3)} & 0 & 0 & 0 & 0 \\ 0 & \frac{0.1}{(32)(57.3)} & 0 & 0 & 0 \\ 0 & 0 & \frac{0.1}{(32)(57.3)} & 0 & 0 \\ 0 & 0 & 0 & \frac{0.04}{32} & 0 \\ 0 & 0 & 0 & 0 & \frac{0.04}{32} \end{bmatrix} \tag{5.19}$$

The Kalman filter design then used

$$Q_o = G_{w_{MIMO}} G^T_{w_{MIMO}} \tag{5.20}$$

$$R_f = G_{n_{MIMO}} G^T_{n_{MIMO}} \tag{5.21}$$

The H_2 problem is formulated as

$$\dot{x}_p = \overbrace{\begin{bmatrix} A_p \end{bmatrix}}^{A_2} \begin{bmatrix} x_p \end{bmatrix} + \overbrace{\begin{bmatrix} \Gamma\sqrt{Q_o} & 0 \end{bmatrix}}^{B_w} \begin{bmatrix} w_A \\ w_B \end{bmatrix} + \overbrace{\begin{bmatrix} B_p \end{bmatrix}}^{B_{u_2}} \delta \tag{5.22}$$

$$\begin{bmatrix} z_A \\ z_B \end{bmatrix} = \overbrace{\begin{bmatrix} H \\ 0 \end{bmatrix}}^{C_z} \begin{bmatrix} x_p \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}^{D_{zw}} \begin{bmatrix} w_A \\ w_B \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ \sqrt{R_c} \end{bmatrix}}^{D_{zu}} \delta \tag{5.23}$$

$$y = \overbrace{\begin{bmatrix} C_p \end{bmatrix}}^{C_{y2}} \begin{bmatrix} x_p \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & \sqrt{R_f} \end{bmatrix}}^{D_{yw}} \begin{bmatrix} w_A \\ w_B \end{bmatrix} + \overbrace{\begin{bmatrix} D_p \end{bmatrix}}^{D_{yu}} \delta \tag{5.24}$$

Appendix B contains the full state space matrices.

5.4.1.2 H_∞ Problem. The sensitivity weighting used for the H_∞ problem was derived by the specifications used by Brown in designing the LQG/LTR controller. This ensures a better one-for-one comparison between the two methods. The target loop transfer function chosen was

$$G(s)K(s) = \frac{20}{s + 0.0001} \tag{5.25}$$

the sensitivity weighting matrix can be written as

$$W_s(s) = \begin{bmatrix} \frac{s+20.0001}{s+0.0001} & 0 & 0 \\ 0 & \frac{s+20.0001}{s+0.0001} & 0 \\ 0 & 0 & \frac{s+20.0001}{s+0.0001} \end{bmatrix} \tag{5.26}$$

the state space representation of this weighting matrix is given by

$$A_{W_s} = \begin{bmatrix} -0.0001 & 0 & 0 \\ 0 & -0.0001 & 0 \\ 0 & 0 & -0.0001 \end{bmatrix} \qquad B_{W_s} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_{W_s} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix} \qquad D_{W_s} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The H_∞ part of the 3x3 MIMO problem is then

$$\begin{bmatrix} \dot{x}_m \\ \dot{x}_s \end{bmatrix} = \overbrace{\begin{bmatrix} A_m & 0 \\ B_sC_m & A_s \end{bmatrix}}^{A_\infty} \begin{bmatrix} x_m \\ x_s \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ B_s \end{bmatrix}}^{B_d} d + \overbrace{\begin{bmatrix} B_m \\ B_sD_m \end{bmatrix}}^{B_{u_\infty}} \delta \tag{5.27}$$

$$e = \overbrace{\begin{bmatrix} D_sC_m & C_s \end{bmatrix}}^{C_e} \begin{bmatrix} x_m \\ x_s \end{bmatrix} + \overbrace{\begin{bmatrix} D_s \end{bmatrix}}^{D_{ed}} d + \overbrace{\begin{bmatrix} D_sD_m \end{bmatrix}}^{D_{eu}} \delta \tag{5.28}$$

$$y = \overbrace{\begin{bmatrix} C_m & 0 \end{bmatrix}}^{C_{y\infty}} \begin{bmatrix} x_m \\ x_s \end{bmatrix} + \overbrace{\begin{bmatrix} I \end{bmatrix}}^{D_{yd}} d + \overbrace{\begin{bmatrix} D_m \end{bmatrix}}^{D_{yu}} \delta \tag{5.29}$$

The order of the H_∞ problem is 11. The values in the state space matrices are given in Appendix B.

5.4.2 5x3 MIMO Example. This set-up uses all five measurements of the system. It is very similar to the 3x3 MIMO set-up. The H_2 design is exactly the same as the 3x3 MIMO problem. The same values for H, Q_c, R_c, Q_o and R_f were used. The only difference is that the C matrix for the plant now has five measurements.

5.4.2.1 H_∞ Problem. The most significant difference between the 5x3 MIMO design and the 3x3 MIMO design is the weighting matrix used. The five outputs make the weighting matrix 5x5, but the requirements for tracking haven't changed. There is no requirement to track pitch and yaw rates (q and r); therefore, the sensitivity weights for q and r are set to one. These measurements will then be used for margin control only and not to improve the tracking. The weighting matrix W_s is then

$$W_s(s) = \begin{bmatrix} \frac{s+20.0001}{s+0.0001} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{s+20.0001}{s+0.0001} & 0 \\ 0 & 0 & 0 & 0 & \frac{s+20.0001}{s+0.0001} \end{bmatrix}$$

this can be represented in state space format as

$$A_{W_s} = \begin{bmatrix} -0.0001 & 0 & 0 \\ 0 & -0.0001 & 0 \\ 0 & 0 & -0.0001 \end{bmatrix} \qquad B_{W_s} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C_{W_s} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix} \qquad D_{W_s} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The set-up for the H_∞ problem is symbolically the same as the 3x3 MIMO problem. The weight changed but the order of the H_∞ problem remained 11 since the two weights added were static. The state space matrices with numerical values for this problem are located in Appendix C.

5.5 Mixed H_2/H_∞ Controller Results

The optimization of this problem is more difficult than the SISO problem due to the number of design variables. The SISO problem, at the order of the H_2 problem, has 23 design variables. The MIMO 3x3 case has 63 design variables. The MIMO 5x3 case has 79 design variables. The increase in the number of design variables from the 3x3 to the 5x3 is caused

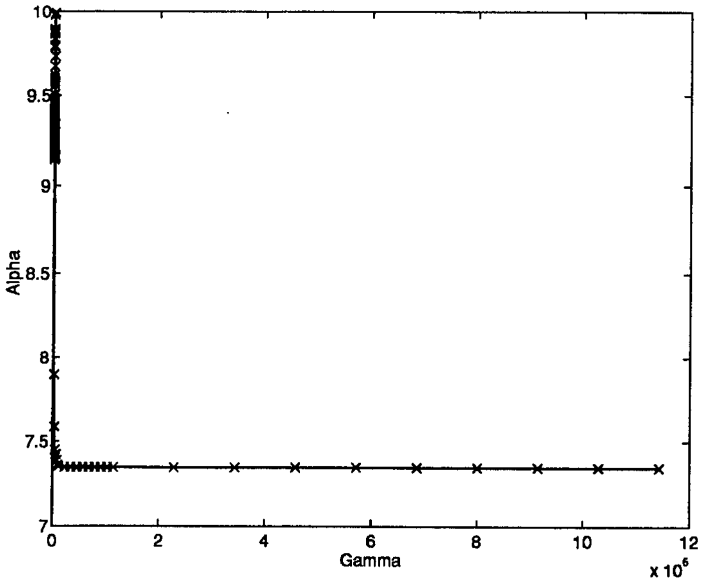

Figure 5.2 Mixed H_2/H_∞ Curve (3x3 MIMO)

by the increased size of the controller C matrix due to the two additional measurements. No mixed H_2/H_∞ controllers were able to be found using the original optimization code. Both of these problems have been run using the improved optimization code and full H_2/H_∞ curves were generated for both examples. The mixed curves along with the sensitivity plots are presented in this section.

5.5.1 3x3 Mixed H_2/H_∞ Controller Design. The mixed H_2/H_∞ controller design for this problem was initialized at the H_2 controller. The order of this controller is eight. This 3x3 MIMO example is what will be used to compare to the LQG/LTR design since two measurements were not used for either design case. The mixed H_2/H_∞ curve for this problem is shown in Figure 5.2. The minimum γ achieved for this problem was 202.85. The minimum absolute $\underline{\gamma}$ for the problem with an 11th order controller is 1.783. The minimum γ at 8th order was expected to be much closer to 1.783. This could be due to two factors; 1) the optimization routine is stuck in a local minimum of the design space and can not move away from that point or 2) the MATLAB 'CONSTR' can not handle the numerics of the problem at that point in the mixed H_2/H_∞ design space.

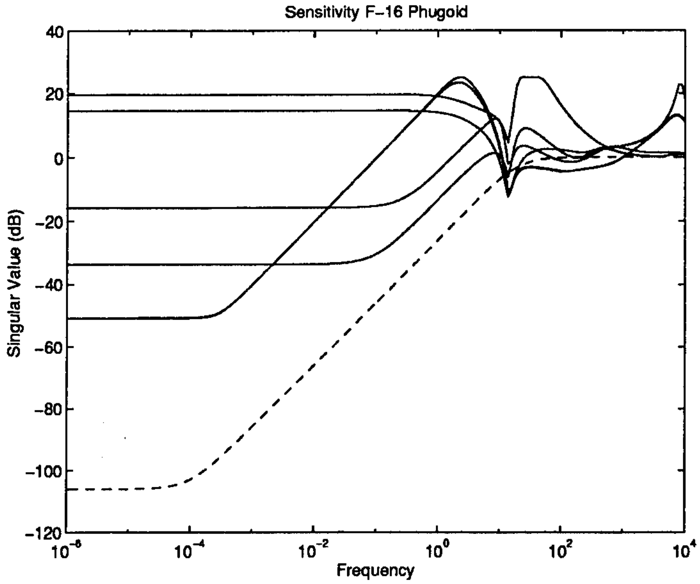

Figure 5.3 Singular Value Plot (3x3 MIMO)

The singular value plot of the closed-loop sensitivity approaches the target sensitivity as γ approaches $\underline{\gamma}$ for the system. This is shown in Figure 5.3. The controller with γ=202.85 produced approximately the same shape for the sensitivity plot as the inverse of the weight, but the plot was shifted up by about 30 dB. This indicates the tracking performance will be less than desired. Also, higher values of γ produced sensitivity plots closer to the target sensitivity.

5.5.2 5x3 Mixed H_2/H_∞ Controller Design. The mixed H_2/H_∞ controller design for this problem was started at the H_2 controller. The order of this controller is eight. This design contains all five outputs. The mixed H_2/H_∞ curve for this problem is shown in Figure 5.4. The minimum γ achieved for this problem was 346.

The maximum singular value of the sensitivity approaches the target sensitivity as γ nears the optimal value for the system. This is shown in Figure 5.5. The controller with an γ equal to 346 produced the sensitivity curve that has the same shape as the target sensitivity, but is shifted up by about 30 dB.

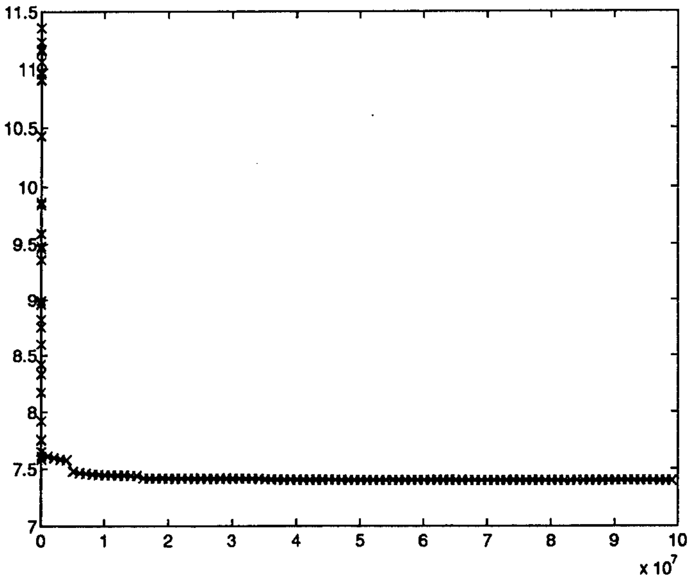

Figure 5.4 Mixed H_2/H_∞ Curve (5x3 MIMO)

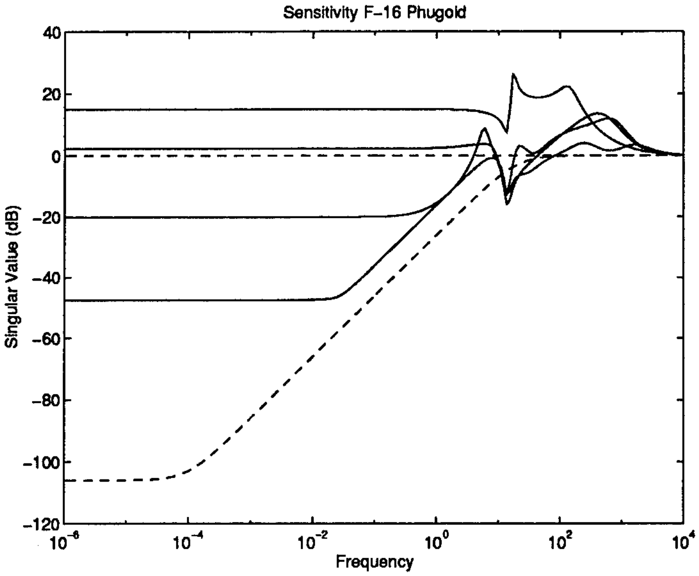

Figure 5.5 Singular Value Plot (5x3 MIMO)

5.6 Comparison of Mixed H_2/H_∞ controller designs to LQG/LTR

The mixed H_2/H_∞ controller with a γ=202.85 was simulated for a lateral acceleration command of 1 G. The simulation showed the tracking performance to be very poor. A simulation of the three outputs for a 1 G step in N_y is shown in Figure 5.6. The LQG/LTR

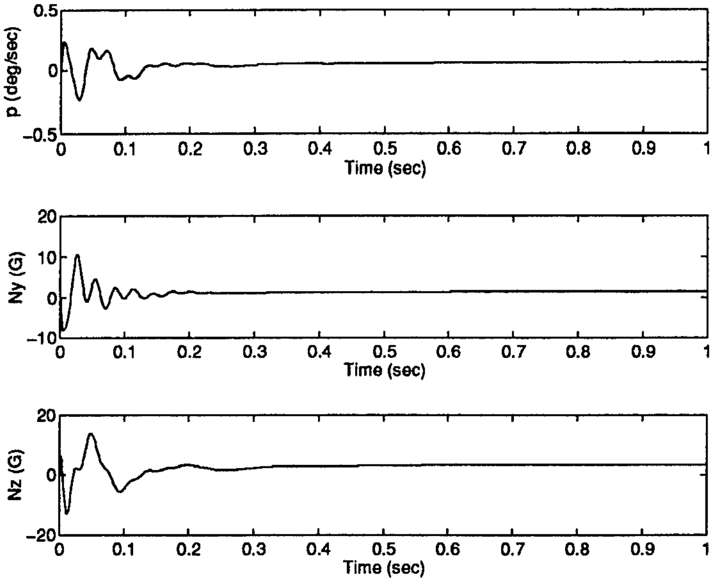

Figure 5.6 Response of system to 1 G N_y step (Mixed H_2/H_∞ Controller)

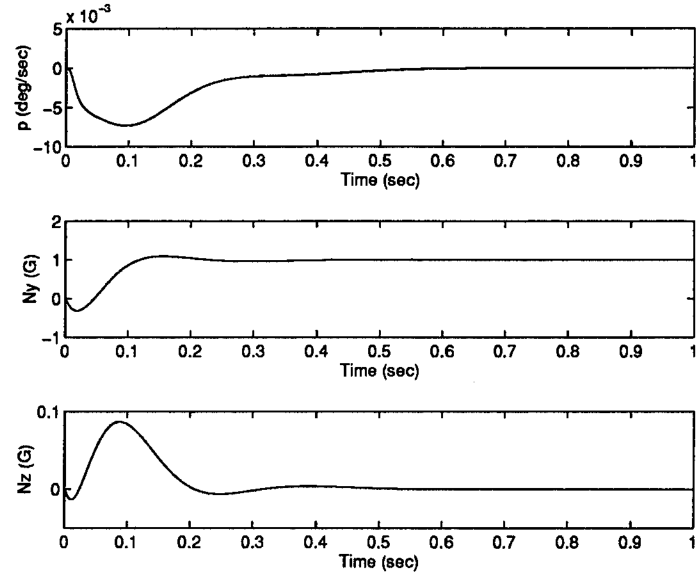

Figure 5.7 Response of system to 1 G N_y step (LQG/LTR Controller)

controller with K_{stat} design for the same step input produced very good tracking, as shown in Figure 5.7.

5.6.1 Comparison of Mixed H_2/H_∞ with K_{stat} to LQG/LTR. The mixed H_2/H_∞ design was done for the original missile plant with the non-minimum phase zeros and the

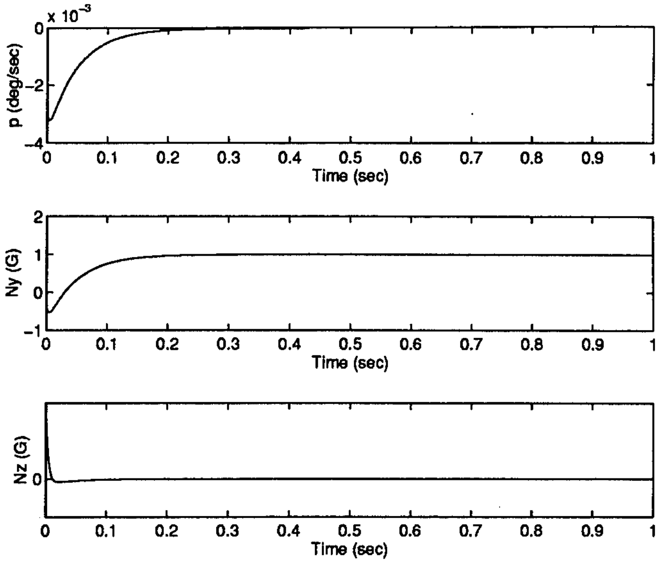

Figure 5.8 Response of system with K_{stat} to a 1 G N_y step (H_∞ Optimal Controller)

lightly damped poles. This is a very hard plant to control. Brown used the static gain matrix to move the poles of the system. Brown's study showed that without moving the poles, the LQG/LTR controller did not perform well. Therefore, the next step was to check the H_∞ optimal controller design for the modified plant, (i.e., with the static gain closed in a loop with the plant). The simulation of the controller showed very good tracking performance. This is shown in Figure 5.8. The optimal H_∞ controller order, however, was eleven. The H_2 optimal order for the problem was eight.

The mixed H_2/H_∞ design was then done for the modified missile plant using K_{stat}, and still for order eight. The mixed H_2/H_∞ curve is shown in Figure 5.9. It was simulated for the same 1 G step acceleration in N_y. The tracking performance was about the same as the design for the original missile plant. There is an unacceptable amount of overshoot and extremely long settling time for the response to a 1 G N_y step. A plot of the response is in Figure 5.10.

5.6.2 Comparison of Mixed H_2/H_∞ at 11th order to LQG/LTR. Previous results showed that the H_∞ optimal controller performed well. See Figure 5.8. The final attempt made to design an H_2/H_∞ controller was to run the optimization code at 11th order (i.e., the

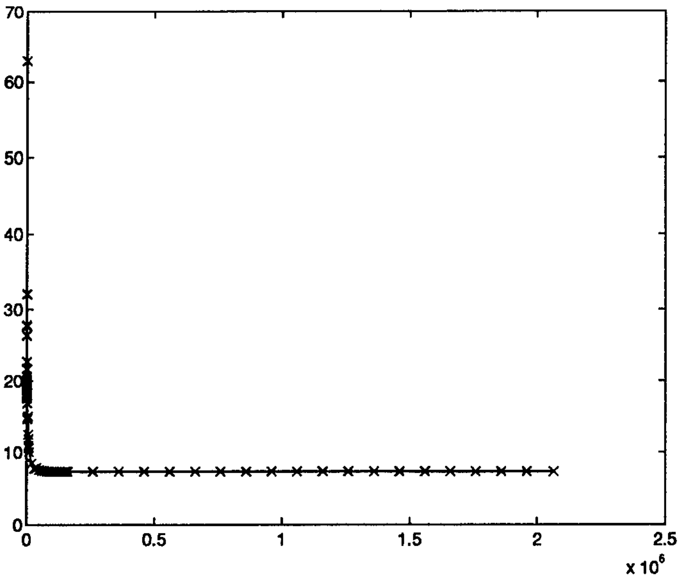

Figure 5.9 Mixed H_2/H_∞ Curve with K_{stat} (3x3 MIMO)

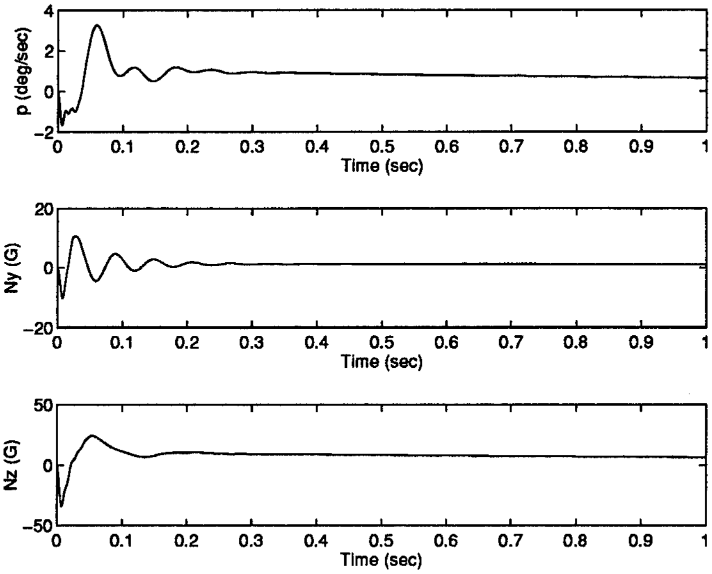

Figure 5.10 Response of Mixed H_2/H_∞ Controller to 1G Ny step with Kstat (H_2/H_∞ Controller)

order of the H_∞ problem). Since using K_{stat} to do the mixed H_2/H_∞ design at 8th order did not improve tracking performance, K_{stat} was not incorporated into the 11th order design. At 11th order, $\underline{\gamma}$=1.76. Therefore, it is known that the optimization routine should be able to get very close to $\underline{\gamma}$=1.76 at this order. The H_2 optimal mixed controller for the 3x3 problem was increased from 8th to 11th order by adding a lag to each transfer function. The lag can be represented as

$$A_{lag} = \begin{bmatrix} -1000 & 0 & 0 \\ 0 & -1000 & 0 \\ 0 & 0 & -1000 \end{bmatrix}$$

$$B_{lag} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad C_{lag} = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

$$D_{lag} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The mixed H_2/H_∞ curve at 11th order requires 87 design variables. This higher order problem controller design was attempted using the MATLAB 'CONSTR' optimization routine. Although the previous 3x3 and 5x3 problems at H_2 order (8th order) could be run using the MATLAB optimizer, the 3x3 problem at 11th order could not be done using MATLAB. No mixed H_2/H_∞ controllers could be found using MATLAB. Next, a FORTRAN/IMSL optimization routine was used for this problem. A FORTRAN program was written that called the same MATLAB files used for the previous examples only where the MATLAB program had called the 'CONSTR' optimization routine the FORTRAN program now called IMSL/DNCONG. For further explanation on the details of MATLAB versus IMSL's optimization routines, see Chapter 8. This routine was able to find mixed H_2/H_∞ designs for this problem. The mixed H_2/H_∞ curve is shown in Figure 5.11. The local minimum in the plot could not be removed by the means discussed in Chapter 6. The lowest γ found

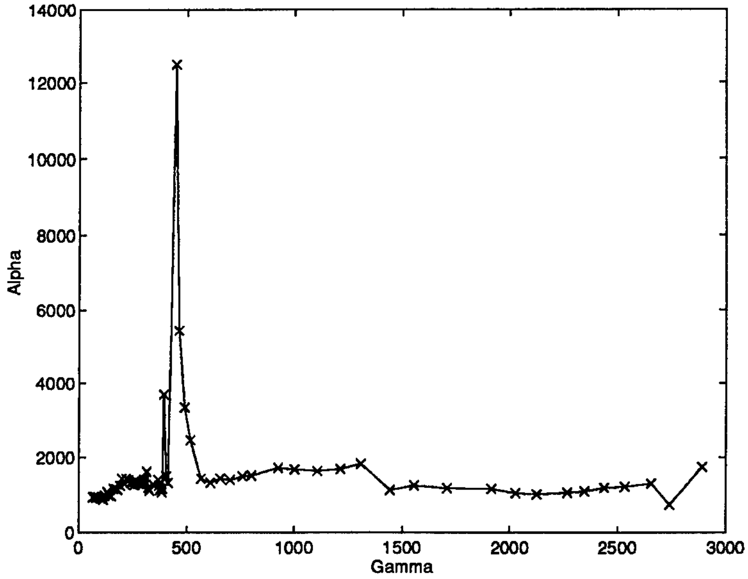

Figure 5.11 Mixed H_2/H_∞ Curve 3x3 MIMO (Order 11)

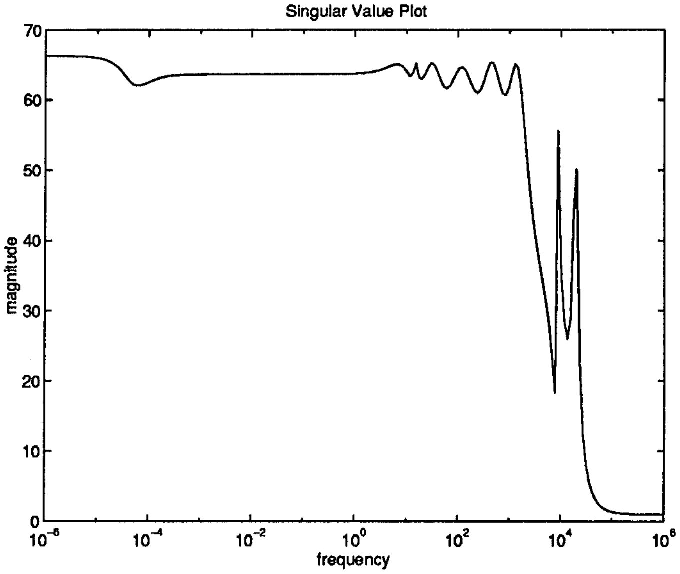

Figure 5.12 Singular Value Plot Mixed H_2/H_∞ 11th Order (γ=66

using the optimization routine was γ=66. The absolute $\underline{\gamma}$ for this problem was 1.76. The difference between these two numbers was much less than for the 8th order mixed design. The closed-loop singular value plot at γ=66 is shown in Figure 5.12. Note how flat the singular value plot has become.

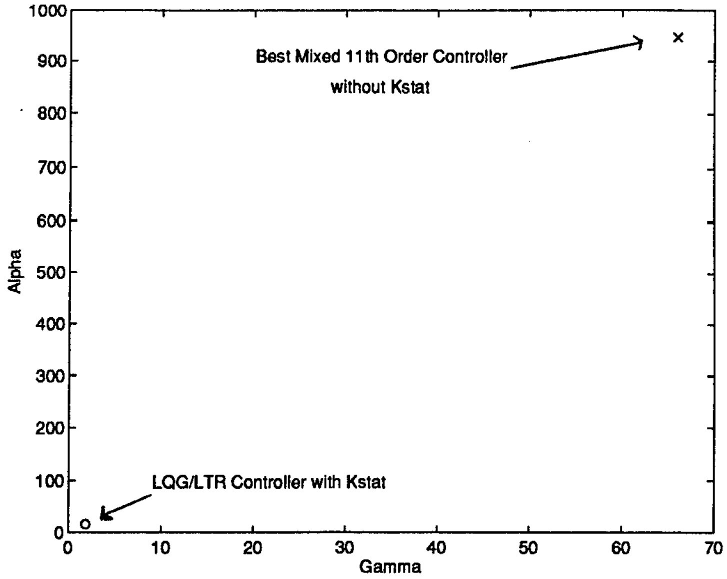

Figure 5.13 Comparison of LQG/LTR Controller to Mixed H_2/H_∞ Controller

A comparison was then made between the α and γ for the LQG/LTR design by Brown with K_{stat} to that found for the mixed H_2/H_∞ controller with K_{stat} at 11th order with the lowest γ found. This is shown in Figure 5.13. This comparison shows two things: 1) that the mixed H_2/H_∞ design can never achieve the α and γ produced by the LQG/LTR controller or 2) that the mixed H_2/H_∞ controller at γ=66 is at a local minimum and could be reduced further. The latter is believed to be the case.

The step response for the mixed H_2/H_∞ controller to a 1 G step in N_y improved as shown in Figure 5.15. However, when compared with the LQG/LTR design, the mixed has five times as much overshoot as LQG/LTR.

5.7 Summary

The mixed H_2/H_∞ controller design was completed for both the 3x3 MIMO and the 5x3 MIMO 8th order problems using the MATLAB optimizer 'CONSTR'. The optimization code was able to find mixed controllers for this larger MIMO example. Mixed controllers can be found in approximately a day for this size of problem(79 design variables). The 11th order could not be optimized using MATLAB 'CONSTR'. Rather, a FORTRAN/IMSL optimizer

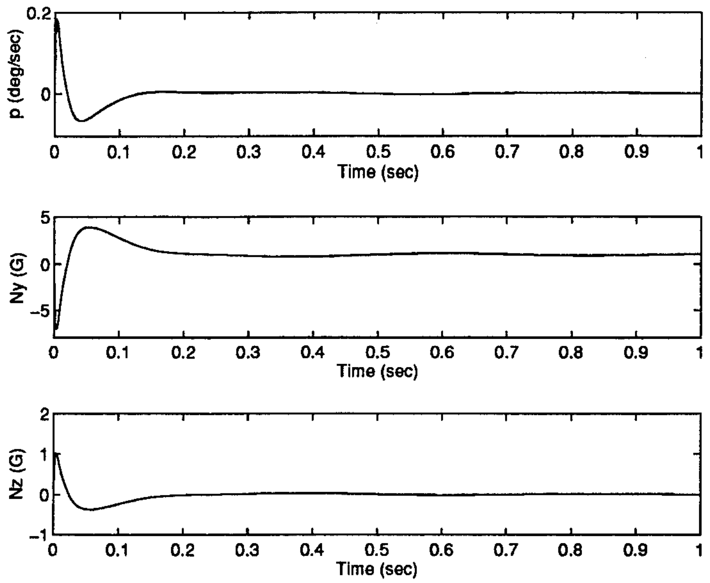

Figure 5.14 Response of system to 1 G N_y step (Mixed H_2/H_∞ Controller Order 11)

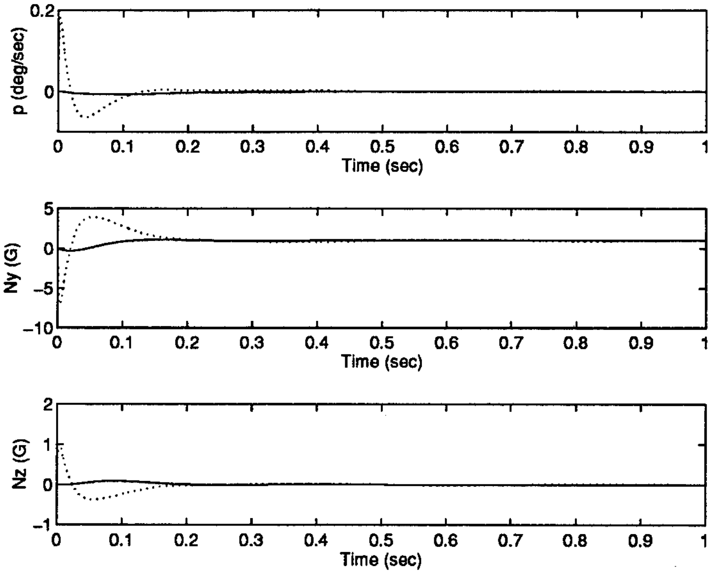

Figure 5.15 Response of system to 1 G N_y step (Mixed H_2/H_∞ Controller Order 11) versus LQG/LTR

was used to generate the mixed H_2/H_∞ curve. Due to the size of the problem, the 11th order problem took several days to converge to γ=66.

Simulation of these controllers shows that the controller does not track a step input very well at 8th order. There is an unacceptable amount of overshoot and the settling time does not meet the requirements. When the optimal order H_∞ controller was tested on this problem, it performed very well, but it was three orders higher than the H_2 problem. The 11th order mixed H_2/H_∞ controller at γ=66 demonstrated improved tracking performance compared to 8th order but was still not as good as LQG/LTR. This is believed to be a local minimum problem. Work is currently being done to help the numerical code "kick out of" local minima. Finally, it was shown that moving the poles of the system did not improve the mixed H_2/H_∞ controller performance as it did for the LQG/LTR design.

VI. Evidence of Local Minimum

6.1 Introduction

It is usually very difficult, when applying optimization techniques, to be sure a global minimum has been found. This is true for any problem that is not convex. The necessary conditions and a positive definite Hessian matrix ensure that the design found by the optimization routine is a local minimum. If the problem is not convex, the only way to ensure that the design is a global optimum is to prove that the Hessian is positive definite for all possible values of design variables. This is rarely practical in a design application. Posing the optimization problem as a convex problem (if possible) is the best way to ensure a design is a global minimum.

The optimal-order mixed H_2/H_∞ controller design was posed as a convex problem by Walker [Wal94]. He states that if the transfer function T_{zw} is in the space H_2 then the two-norm of $T_{zw}(Q)$ is a strictly convex functional of Q. If a function is convex there is one global minimum and therefore the design at that point is unique. The optimal order for the mixed problem has not be determined analytically. Numerical results have suggested that the optimal order is larger than the combined H_2 and H_∞ plant. Wells and Ridgely [WR92] have suggested that the optimal order is infinite. The optimal-order mixed H_2/H_∞ controller design is a convex problem in the design space of the Youla parameter Q, but what about the fixed-order controller problem? The two-norm objective and the infinity-norm and stability constraints are not convex functions in the design variable space. Currently, it is not know whether it is possible to pose the fixed-order mixed problem in a convex setting.

This section present results for both the F-16 SISO problem and the missile MIMO problem that show evidence that the fixed order problem is not convex as posed in this work. There are mixed H_2/H_∞ curves generated for both these examples which show that local minimum occur.

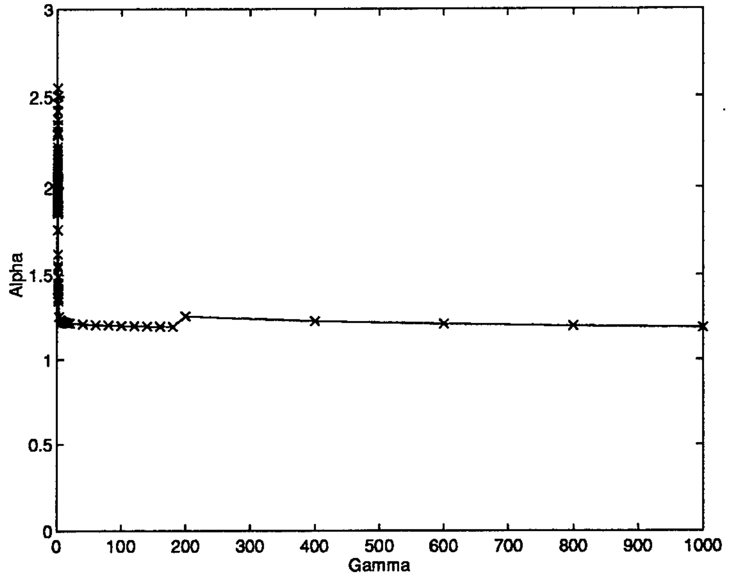

Figure 6.1 Mixed H_2/H_∞ Curve Local Minimum (H_2 Order)

6.2 Examples of Local Minimum

Both the F-16 and missile examples will be discussed in this section. The local minimum are more frequent and more pronounced in the missile example.

6.2.1 F-16 Example. The F-16 example was discussed in Chapter IV. This example was the first to show evidence of a local minimum. The local minima for this example occur more frequently when small steps in γ are taken. They also occur more frequently the closer the design gets to H_∞ optimal. Figure 6.1 shows a place in the mixed H_2/H_∞ curve where a local minimum occurred.

Further evidence that this point was a local minimum was the singular value plot. At the point the local minimum occurred, the shape of the singular value plot changed drastically. This lead to the conclusion that a completely different controller design had been found, and that a local minimum was the reason for this design change. Figure 6.2 shows the change in the singular value plot.

6.2.2 Missile Example. The missile MIMO example also demonstrated local minima. The trends are similar to the F-16 case, in that they occur more frequently with small

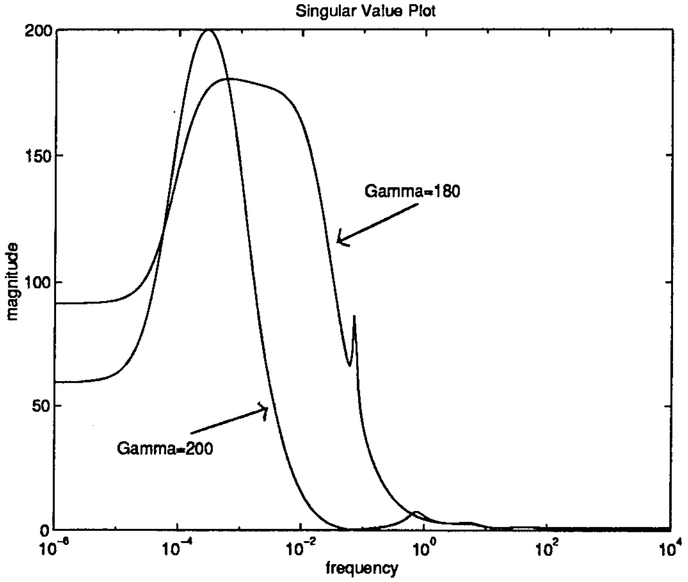

Figure 6.2 Singular Value Plot where local minimum occurred (H_2 Order)

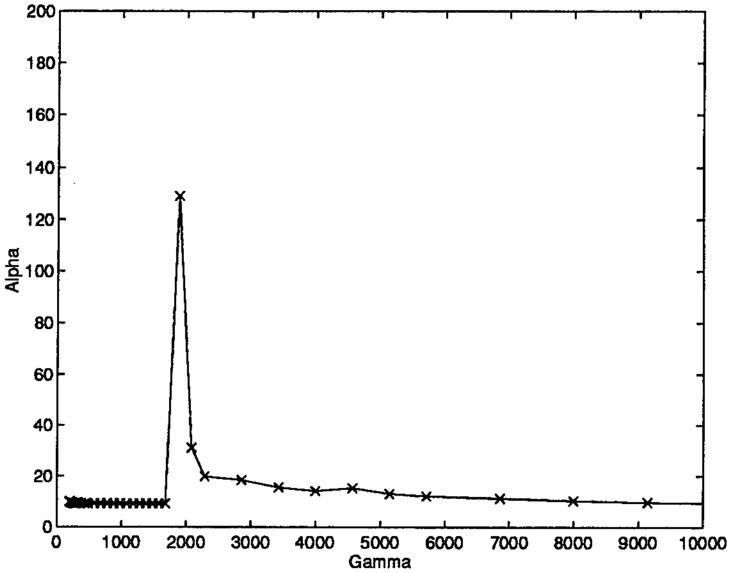

Figure 6.3 Mixed H_2/H_∞ Curve Local Minimum (3x3)

steps in γ. They also occur more frequently the closer the mixed design gets to H_∞ optimal. A plot with local minima for the MIMO 3x3 example is shown in Figure 6.3.

The MIMO 5x3 example also showed local minimum. A plot of the mixed H_2/H_∞ curve for this example is in Figure 6.4. An example of local minimum occurring very close to optimal is shown in Figure 6.5.

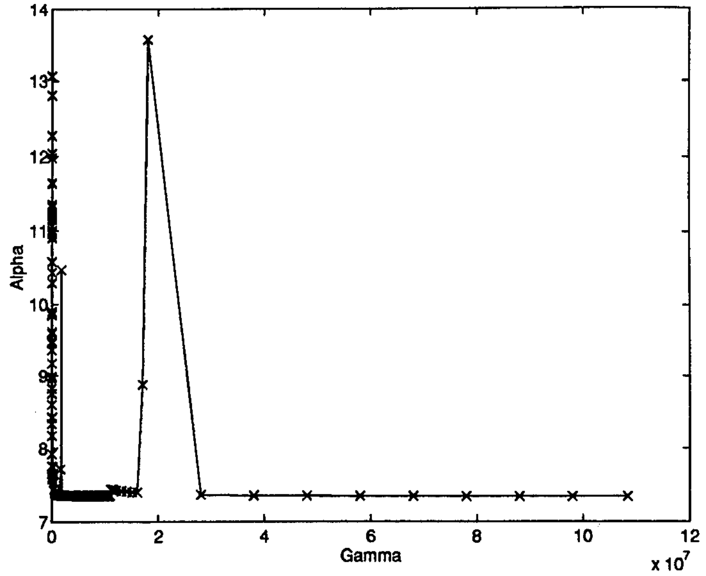

Figure 6.4 Mixed H_2/H_∞ Curve Local Minimum (3x5)

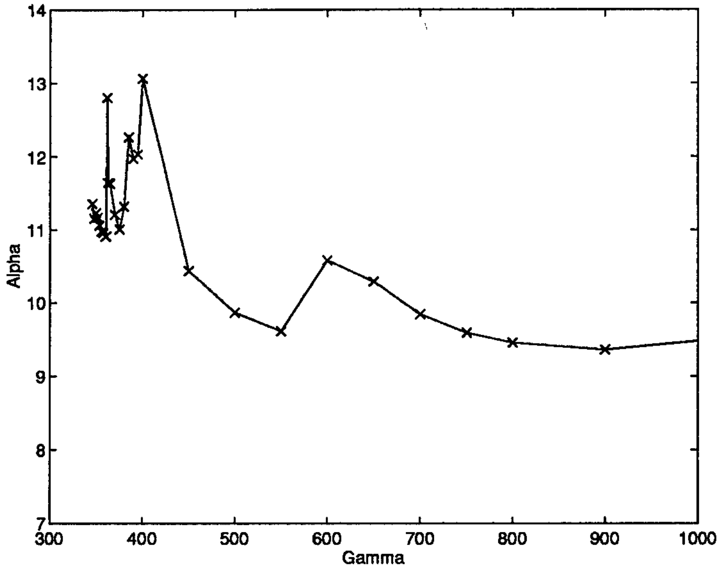

Figure 6.5 Mixed H_2/H_∞ Curve Local Minimum (3x5)

6.3 Solutions to Local Minimum

One way to prevent local minima from occurring is by taking large steps in γ, thereby possibly skipping over a local minimum. This is not a good solution, however, since large steps may cause numerical ill conditioning. Smaller steps work well to make sure low points in the design space are found. The drawback of small steps is that there is more of a chance the optimizer will be stuck in a local minimum and will not be able to move enough to find a lower local minimum or the global minimum. Methods such as simulated annealing offer the ability to move away from a local minimum and possibly find a global minimum. Although they were not used for this study, they could possibly decrease the chances of the design getting stuck in a local minimum point when taking small steps.

Once a local minimum has been found, there is a way to "fix" the mixed H_2/H_∞ design curve. This is done by using the controller that was found just after the drop in the mixed H_2/H_∞ curve. Using this controller, the drop in the mixed H_2/H_∞ curve can be "removed" by tracing the curve in the opposite direction, (i.e., taking positive steps in γ). This method was used to remove the local minimum points for the F-16 and missile examples. This is the reason the examples in Chapters 4 and 5 do not show the local minimum points as described in this chapter.

6.4 . Summary

Local minimum have occurred in both the F-16 and missile example completed for this thesis. The reason these local minimum have not been obvious before is due to the limited reliability of the code. When only seven to ten controllers could be calculated for the entire mixed H_2/H_∞ curve, the chances of seeing this problem were very low. With the improved reliability and the means to generate hundreds of controllers for a specific example, the problem has become more evident.

These examples demonstrated that the chances of a local minimum becoming obvious are increased when smaller step sizes in γ are taken. Local minima also appear more frequently the closer the design gets to H_∞ optimal. There is no means to determine if the curve generated is the absolute lowest local minimum. However, if a local minimum is obvious to the designer, (i.e., the mixed curve drops to a lower part of the curve), that local minimum point can be removed from the mixed curve. This clearly shows that the fixed order problem is not convex in the design variable space.

VII. Time Savings Comparisons

The main objective of this thesis was to reduce the time to compute a mixed H_2/H_∞ controller. To verify that this was done, a time comparison was made between the old software and the new improved software for two examples. This was done for the small SISO system discussed in Chapter 3 and for the F-16 problem. No time comparison could be made for the missile problem, since no solution could be reached with the original code.

7.1 SISO Second Order System

Table 3.1 showed the number of required function evaluations. This section will present a comparison of the required amount of CPU time in seconds for the example to be run on a dedicated SPARC20 SUN workstation using the old and new optimization code.

Table 7.1 shows the second order SISO example discussed in Chapter 3 with the old code and the peaks manually divided. This comparison was chosen since a complete curve could not be generated without the peaks constrained separately. Note, as discussed in Chapter 3, the original code with separately constrained peaks had five out of the fourteen cases where the iteration terminated by maximum iterations exceeded. This means the tolerances set on the design variables, the function and the constraints were not met. Therefore, the code would have taken even longer to run if the limit on maximum number of iterations was not imposed.

7.2 F-16 Example

The original code was run for the F-16 example. Few points were generated due to the time required to generate one point. A comparison of the time to generate those points using the old code versus the new code is shown in Table 7.2. To get an answer, the scale factors of the original code were set at λ_1=0.001, λ_2=1 and λ_3=1000. For other scale factors, such as 0.01, 1 and 10 respectively, a controller for the first γ step could not be found.

Table 7.1 Time Comparison of Old versus New Optimization Code (Test Example)

	OLD CODE	NEW CODE
γ	CPU sec	CPU sec
14	436.36	8.16
13	1795.41	3.39
12	2130.45	3.38
11	4895.68	3.58
10	4835.21	3.69
9	4903.26	6.8
8	1949.76	3.41
7	1974.93	4.25
6	167.21	3.98
5	521.26	4.07
4	4816.35	4.05
3	464.28	4.71
2	5224.44	4.72
1	3238.2	85.58
TOTAL	37352.8	143.71

Table 7.2 Time Comparison of Old versus New Optimization Code (F-16 Example)

	OLD CODE	NEW CODE
γ	CPU sec	CPU sec
76744	518.10	17.21
75744	31.45	11.91
74755	91.80	14.01
TOTAL	641.35	43.13

7.3 Summary

It is obvious that improving the gradients alone for this problem would increase the reliability and efficiency of the code. The tables in this chapter give a perspective of just how much time is saved from the original method compared to the new optimization code. Examples which originally took hours now take minutes.

VIII. MATLAB Compared to FORTRAN IMSL/DNCONG

The purpose of this chapter is to compare MATLAB to FORTRAN IMSL/DNCONG for mixed H_2/H_∞ optimization. Both MATLAB and the FORTRAN IMSL routines use SQP as the method of optimization. However, the SQP method is implemented differently in each routine. First, a background on IMSL and its implementation of SQP is given. Next, a background on the MATLAB optimization package and its implementation of SQP is presented. Following the background of these methods, a comparison between the SQP implementations is made. Finally, the F-16 example is used to compare run times and convergence between MATLAB and FORTRAN IMSL.

8.1 Background IMSL Optimizer

The IMSL FORTRAN library contains optimization routines classified in the following categories: unconstrained minimization, minimization with simple bounds, linearly constrained minimization, and nonlinearly constrained minimization. The routines are proprietary and can not be altered by the user. The DNCONG routine in that library solves a general nonlinear programming problem using the SQP algorithm with a user supplied gradient. It is a double precision routine. The optimization algorithm is based on a FORTRAN code developed by Schittkowski [Sch85]. First, the general optimization problem is stated.

$$\begin{array}{rll} \text{minimize} & f(x) & \\ \text{subject to:} & & \\ \text{equality constraints:} & g_j(x) = 0 & j = 1, ..., m_e \\ \text{inequality constraints:} & g_j(x) = 0 & j = 1 + m_e, ..., m \\ \text{side constraints:} & x_1 \leq x \leq x_u & \end{array}$$

8.1.1 IMSL Implementation of Quadratic Sub-Problem. The IMSL routine includes two improvements that make the code more efficient. The first improvement is the use of an active set strategy to calculate constraint gradients. This reduces the number of gradient calculations required. The active set flag was not utilized for this thesis; therefore, this did not effect the comparison between IMSL and MATLAB.

The second improvement in IMSL's implementation of SQP is the introduction of a variable δ which is added to the subproblem to minimize possible inconsistent subproblems. This equation is only used if a search direction cannot be found for $\delta = 0$. The QP problem given by (2.104) through (2.106) is rewritten here as

$$\text{minimize } s^T \nabla f + \frac{1}{2} s^T W s + \frac{1}{2} p_k \delta^2 \tag{8.1}$$

$$\text{subject to} \tag{8.2}$$

$$(1-\delta) g_j(x_k) + s^T \nabla g_j(x_k) \begin{Bmatrix} = \\ \geq \end{Bmatrix} 0, j \in J_k \tag{8.3}$$

$$g_j(x_k) + s^T \nabla g_j(x_{k_j}) \geq 0, j \in K_k \tag{8.4}$$

$$x_l - x_k \leq s \leq x_u - x_k \tag{8.5}$$

$$0 \leq \delta \leq 1 \tag{8.6}$$

where x is a vector of design variables and H is updated using a quasi-Newton method to approach the Hessian matrix. The update method used in IMSL is the BFGS method discussed in Chapter 2.

8.1.2 IMSL Implementation of the Merit Function. The merit function is used to determine the end of a line search. The next values of the design variables and the Lagrange multipliers are defined at the end of the line search. The equations for the design variables (x_{k+1}) and Lagrange multipliers (v_{k+1}) are

$$x_{k+1} = x_k + \beta_k s_k \tag{8.7}$$

$$v_{k+1} = v_k + \beta_k (u_k - v_k) \tag{8.8}$$

where u_k is the Lagrange multiplier used in the quadratic sub-problem.

The merit function is minimized so that the design vector x_{k+1} is in the feasible design space and the design vector shows some improvement. The penalty function used in IMSL is

$$\phi_k(\alpha) = \varphi_{r_k}\left(\begin{pmatrix} x_k \\ v_k \end{pmatrix} + \beta\begin{pmatrix} s_k \\ u_k - v_k \end{pmatrix}\right) \tag{8.9}$$

where φ is a penalty function and r_k is a vector of penalty parameters. The IMSL routine implements the augmented Lagrangian function proposed by Schittkowski as the penalty. This method has been determined to be more efficient than an exact penalty function. The augmented Lagrange penalty is

$$\varphi_r(x,v) = f(x) - \sum_{j=1}^{m_e}(v_j g_j(x) - 0.5 r_j g_j^2(x)) - \sum_{j=m_e+1}^{m'}\left\{\begin{matrix} (v_j g_j(x) - 0.5 r_j g_j^2(x)) & if\ g_j(x) \leq v_j/r_j \\ 0.5 v_j^2/r_j & otherwise \end{matrix}\right\} \tag{8.10}$$

where $m' = m + 2n$ is used to include the design variable bounds as constraints. The line search starts at $\beta = 1$ and is reduced until (8.11) is met, where $\varphi_k'(0)$<0 must hold.

$$\phi_k(\alpha) \leq \varphi_k(0) + \mu\alpha\varphi_k'(0) \tag{8.11}$$

where μ is a constant and φ_k<0. The penalty parameters are used to guarantee convergence. There is a different penalty parameter for each constraint, given by

$$r_j^{(k+1)} = \max\left(\sigma_j^{(k)} r_j^{(k)}. \quad \frac{2m(u_j^{(k)} - v_j(k))^2}{(1-\delta_k)s_k^T W s_k}\right),\ \text{j=1,...m} \tag{8.12}$$

where

$$\sigma_j^{(k)} = \min\left(1, \quad \frac{k}{\sqrt{r_j^{(k)}}}\right) \tag{8.13}$$

8.1.3 Convergence Criteria. The convergence criteria for IMSL is less restrictive than the criteria for MATLAB. It is not an option to input to the IMSL code. The source code is required to modify this parameter. The following is the convergence criteria on the objective, where KTO is a measure of the Kuhn-Tucker optimality conditions.

$$\text{KTO} = \left|s^T \nabla f\right| + \sum_{i=1}^{m} |u_j g_j(x)| \leq \varepsilon \tag{8.14}$$

The convergence criteria on constraints is measured by SCV the sum of constraint values. This criteria is

$$\text{SCV} \leq \sqrt{\varepsilon} \tag{8.15}$$

The variable ϵ is the error tolerance. This is the variable that the user can set if the source code is available.

8.2 Background MATLAB Optimizer

The MATLAB optimization toolbox contains functions that solve many types of general nonlinear optimization problems. Some examples of optimization problems MATLAB can solve are constrained and unconstrained problems, nonlinear least square problems and matrix minimization problems. MATLAB also allows the user to supply gradients. If gradients are not supplied by the user, finite difference gradients are calculated. The constrained minimization routine in MATLAB 'CONSTR' uses SQP. The MATLAB implementation is discussed in this section.

8.2.1 MATLAB Implementation of Quadratic Subproblem. The MATLAB routine uses a quadratic subproblem to calculate the search direction. It is a quadratic approximation of the Lagrangian with no constant term. The nonlinear constraints have been linearized for this problem. The quadratic subproblem that the MATLAB routine solves is given in (2.104) through (2.106). This sub-problem is solved using a Quadratic Programming (QP) routine in MATLAB. The solution to the QP problem is iterative. The MATLAB QP solver calculates all constraints, decides which ones are active, and then uses those to solve the QP problem. It should be noted that all the constraints and their gradients must be calculated, and then the routine determines which ones are active.

8.2.2 Merit Function. Once the search direction is found the step length in that direction must be determined. A requirement of the step length is that it must decrease a merit function by a certain amount. The MATLAB implementation has two merit functions. The first merit function is an exact penalty that is written as:

$$\phi_1 = f(x) + \sum_{i=1}^{m_e} r_i g_i(x) + \sum_{i=m_e}^{m} r_i \max\{\, 0, \quad g_i(x)\,\} \tag{8.16}$$

where

$$r_{ki} = \max \lambda_i, \frac{1}{2}(r_{(k-1)i} + \lambda_i), \text{i=1,...,m} \tag{8.17}$$

This penalty function allows a contribution from old Lagrange multipliers, which were inactive in the QP solution, but which were recently active. This less stringent merit function was added to improve convergence. The second merit function is used to improve either the constraint or the objective function unless the QP subproblem is infeasible. If the QP problem is infeasible the routine only tries to reduce the maximum constraint. This second merit function is:

$$\phi_2 = \left\{ \begin{array}{ll} \max_j g_j(X) & \text{if X is infeasible} \\ \frac{-1}{F(X)+1} & \text{if X infeasible and F(X) > 0} \\ F(X) - 1 & \text{otherwise} \end{array} \right\}. \tag{8.18}$$

Both of the merit functions (8.16) and (8.18) must be satisfied and the number of iterations in the line search must be less than some maximum for the line search to continue. If a step length of one does not meet the criteria of the two merit functions, the step length is bisected and another function evaluation is done. This is completed up to the number of maximum iterations. Once the step size is less than 10^{-4}, the step length is set negative at every other bisection step. The step length is divided fifteen times before the step length of 10^{-4} is set. This forces the optimizer to look in the opposite direction than the original search direction.

8.2.3 Convergence Criteria. The convergence criteria for the objective, constraint and the design variables for MATLAB can be input by the user. The optimization is considered successful when:

1. The maximum absolute change in the search direction is less than two times the tolerance on the design variables.
2. The gradient of the objective multiplied by the search direction is less than two times the tolerance on the objective.
3. The maximum constraint is less than or equal to a set tolerance from zero (i.e., the maximum constraint must be active). This was modified for the H_2/H_∞ optimization problem for the criteria to be the maximum constraint must be within a set tolerance from zero.

8.3 Comparison of Methods

The IMSL method provides an improved active set strategy to calculate the gradients compared to MATLAB. This means that fewer gradients should be calculated by the user when the active flag is used. The biggest difference in this comparison is that IMSL has an improved merit function and line search methodology compared to MATLAB. Instead of a bisection of the step length as in the MATLAB routine, the IMSL interpolation is limited by default to, at most, five function evaluations per line search. The penalty function used in the merit function in MATLAB can reduce the effectiveness of convergence of the problem. Finally, the modified QP problem in IMSL should prevent failure to produce a search direction when constraints are grossly violated. These comparisons lead to the conclusions that IMSL should be more efficient and more accurate in the optimization process than the MATLAB optimization routine. To prove this conjecture, the F-16 SISO example was run using both optimization routines and comparisons were made. This is discussed in Section 8.4.

Table 8.1 Convergence Criteria for MATLAB

Comparison	Objective	Constraint
1	1e-8	1e-4
2	1e-8	1e-4
3	1e-8	1e-4

8.4 Comparison Using F-16 Example

The F-16 example discussed in Chapter 4 was run using the IMSL/DNCONG optimization routine. The FORTRAN program called MATLAB as an "engine" so that MATLAB routines for controls were available. Therefore, the core MATLAB code was called from FORTRAN to generate the function and gradient calculations and the IMSL/DNCONG routine was used for the H_2/H_∞ optimization.

Three comparisons were made for this example. The first was done at H_2 optimal and then reduced in γ increments of 1000. The second comparison was done close to the knee of the curve. It began at a γ value of 100 and reduced γ by steps of 5. Finally, a comparison was made at the knee of the mixed curve where many multiple peaks in the singular value plot were evident. This comparison started at a γ level of 5 and was reduced by γ steps of 0.5. Note the absolute $\underline{\gamma}$ for a seventh order controller is 1.32. The order of the mixed controller for this example is six. These three comparisons will be discussed in the following subsections.

The comparisons were made as close as possible by setting the tolerances in MATLAB as close to the tolerance in IMSL. See Tables 8.1 and 8.2 The numbers were not always exactly the same since the criteria for IMSL convergence on a constraint is measuring the sum of all the violated constraints and comparing that to the tolerance. When multiple peaks were constrained and caused a constraint violation this tolerance was relaxed to provide a better comparison. This provided as close a match for convergence between the two routines as possible. The maximum number of iterations within a line search was also adjusted in IMSL from 5 to 20.

Table 8.2 Convergence Criteria for IMSL

Comparison	KTO	SCV
1	1e-8	1e-4
2	1e-8	1e-4
3	1e-4	1e-2

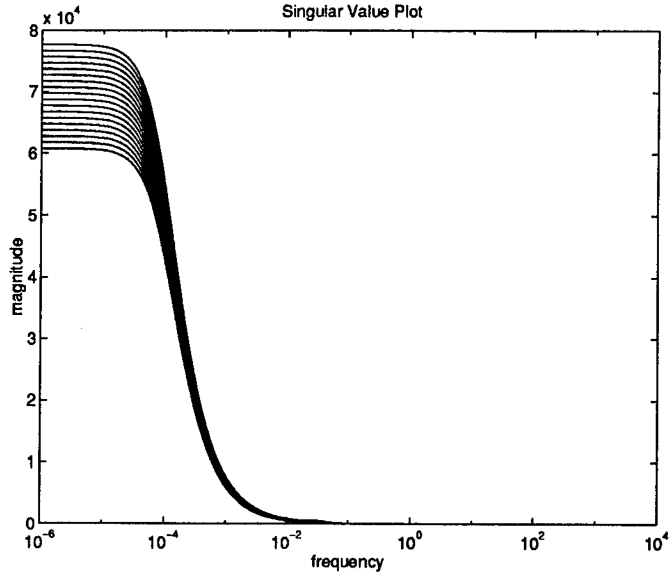

Figure 8.1 Singular Value Plot MATLAB (Comparison 1)

8.4.1 Comparison 1: Initial γ at H_2 Optimal, $\Delta\gamma$=-1000 . This comparison was done to investigate the performance of the two methods beginning at the H_2 optimal compensator. Large steps were taken to identify how the methods handled updating the Hessian matrix. The number of γ steps was set at 70. The MATLAB routine became numerically ill conditioned with Hessian update errors at the 18th γ step. The IMSL routine completed all 70 γ steps. The singular value plots for the MATLAB and IMSL examples are shown in Figures 8.1 and 8.2.

The number of function and gradient evaluations is presented in Table 8.3. Only the first 18 iterations of IMSL are presented for comparison purposes, since that was the number of iterations completed by MATLAB. The IMSL routine used approximately 100 less function and gradient evaluations than the MATLAB optimizer. A comparison of CPU time per each

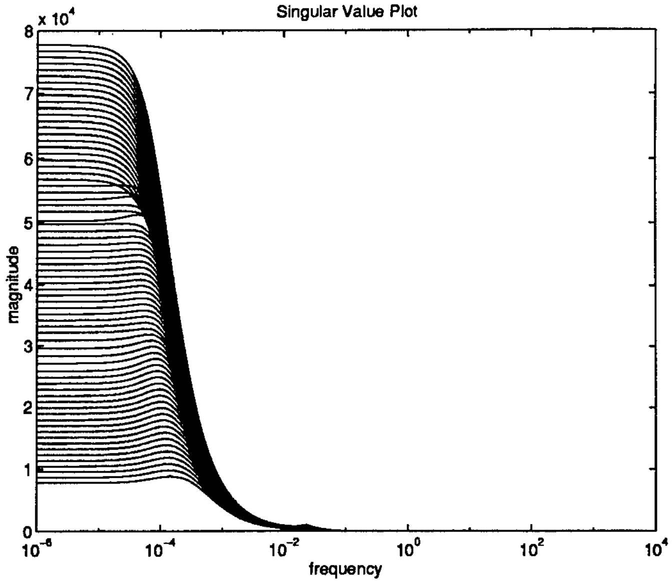

Figure 8.2 Singular Value Plot IMSL (Comparison 1)

step of γ is presented in Table 8.4, where the IMSL and MATLAB times are practically the same. This was true even though the number of function and gradient evaluations was much less for IMSL. This may be due to more time spent in the IMSL optimizer versus time spent in the MATLAB optimizer or due to the extra overhead for IMSL to call MATLAB as an engine. Again, the first 18 steps of the IMSL run are compared here to the MATLAB data.

A comparison of the mixed H_2/H_∞ curves are given in Figure 8.3. The MATLAB curve could not go below γ=60,000. It is difficult to see on the plot, but the MATLAB curve at γ=61744 shows a dip attributed to a local minimum point. This dip causes the MATLAB curve to be located on the IMSL curve. The IMSL mixed curve shows a smooth but non-monotonically increasing plot.

8.4.2 Comparison 2: Initial γ=100, $\Delta\gamma$=-5.

This comparison was done close to the knee of the mixed H_2/H_∞ curve. There was only one peak for most of the iteration so the tolerance used was 10^{-8}.

Both IMSL and MATLAB completed all 19 iterations. The singular value plots of MATLAB and IMSL are given in Figures 8.4 and 8.5. The MATLAB optimizer found a

Table 8.3 Comparison 1 Number of Function and Gradient Evaluation

	MATLAB		IMSL	
γ	NUM FUN	NUM GRAD	NUM FUN	NUM GRAD
76744	13	12	17	15
75744	16	12	15	10
74744	20	14	11	9
73744	11	9	14	8
72744	17	9	14	9
71744	36	19	11	7
70744	14	6	13	8
69744	14	6	12	7
68744	13	6	10	6
67744	37	19	13	7
66744	13	6	19	14
65744	17	11	8	4
64744	11	5	9	5
63744	15	9	9	5
62744	15	10	9	5
61744	32	22	10	5
60744	7	5	11	6
TOTAL	301	280	205	130

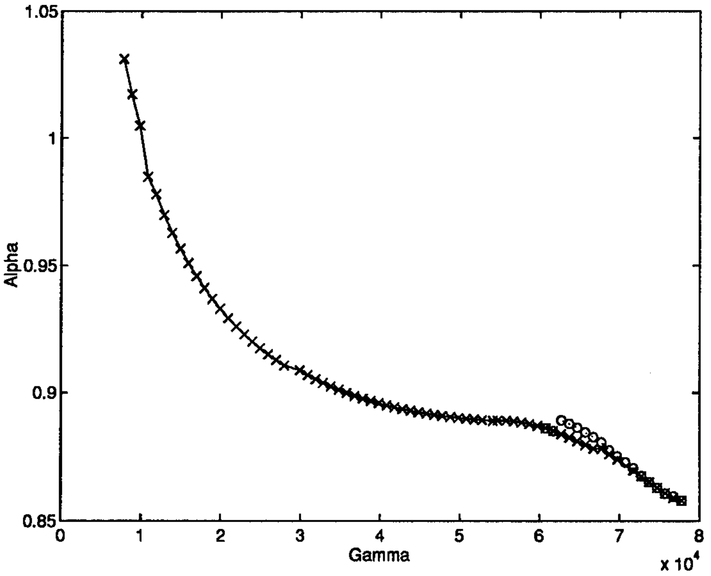

Figure 8.3 Mixed H_2/H_∞ Curve MATLAB-'o' IMSL-'x' (Comparison 1)

Table 8.4 Comparison 1 CPU Time of MATLAB vs IMSL

	MATLAB	IMSL
γ	CPU sec	CPU sec
76744	27.21	31.92
75744	20.36	21.87
74744	21.93	19.00
73744	17.36	19.33
72744	19.26	21.07
71744	28.39	18.47
70744	17.25	20.55
69744	17.01	19.70
68744	16.78	18.36
67744	29.31	21.20
66744	17.53	27.54
65744	20.00	16.34
64744	16.06	17.64
63744	18.69	19.11
62744	19.30	18.55
61744	28.14	19.00
60744	15.21	20.22
TOTAL	349.88	349.74

different design than the IMSL optimizer. The shapes of the singular value plots are different at the first values of γ. The shape of the MATLAB singular value plot indicates a local minimum will be evident on the mixed H_2/H_∞ curve for MATLAB near the first three γ steps. The local minimum for the IMSL run should appear at γ=30.

The number of function and gradient evaluations is presented in Table 8.5. The number of function evaluations for IMSL was 600 more than for MATLAB. This increase in function evaluations can be attributed to not relaxing the tolerance on IMSL enough for the multiple peaks to match the tolerance used for MATLAB. Since the convergence criteria was not the same for MATLAB and IMSL different local minima were found during the optimization process. This is evident from the singular value plots using the two methods. IMSL did not find the lower local minimum as soon as MATLAB did. Also, due to different convergence criteria, the shapes of the singular value plots are different between γ=90 through γ=30. The

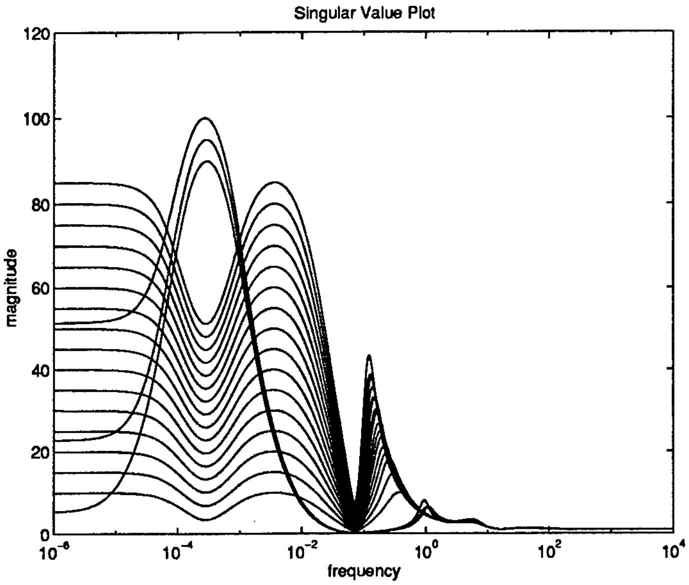

Figure 8.4 Singular Value Plot MATLAB (Comparison 2)

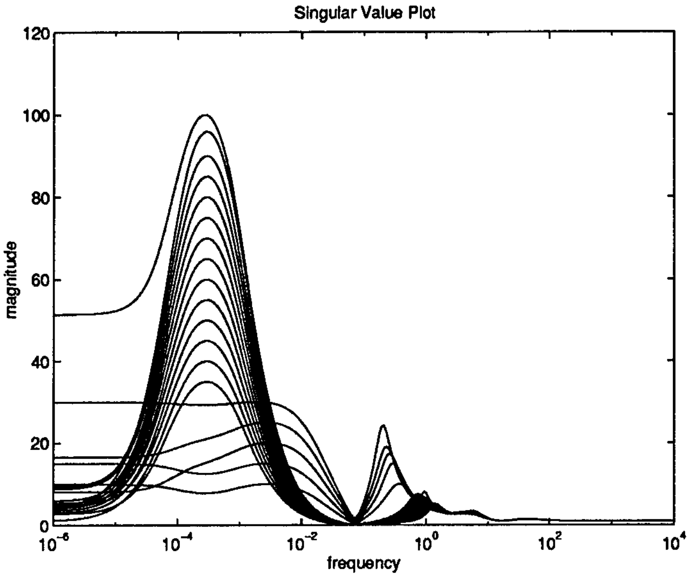

Figure 8.5 Singular Value Plot IMSL (Comparison 2)

time in CPU for both methods is shown in Table 8.6. The increase in time is due to the increase in the number of function evaluations. This increase can also be attributed to the tolerance set for the convergence criteria of the IMSL routine.

The mixed H_2 / H_∞ curves show a significant difference in local minima. The MATLAB optimization routine found a local minimum after three γ steps. The IMSL routine found a

Table 8.5 Comparison 2 Number of Function and Gradient Evaluation

	MATLAB		IMSL	
γ	NUM FUN	NUM GRAD	NUM FUN	NUM GRAD
95	41	20	190	100
90	28	10	13	8
85	178	118	9	7
80	12	8	10	8
75	9	4	9	7
70	10	6	9	7
65	6	6	12	10
60	9	4	23	15
55	8	6	20	12
50	7	4	34	23
45	7	6	17	11
40	11	7	12	8
35	5	4	14	12
30	4	4	186	100
25	4	4	147	81
20	5	5	5	5
15	4	4	7	7
10	8	8	12	10
5	8	8	268	98
TOTAL	364	236	997	529

local minimum after 13 steps in γ. This difference in local minimum can be attributed to the difference in convergence criteria used between the two methods. Both curves do end at the same mixed design points but both took different paths. Also when comparing function evaluations and time required it can be seen that after MATLAB found the lower local minimum it was much quicker in calculating the rest of the curve. IMSL, on the other hand, took longer to find that local minimum and required more function evaluations and time. The MATLAB and IMSL mixed curves are presented in Figure 8.8.

Since it is evident that the IMSL routine was not tracking the same local minimum as MATLAB due to convergence criteria the comparison is not valid. It would be a better comparison for number of function evaluations and time if both MATLAB and IMSL were tracking the same local minimum. Removing the scale factor of (0.0001) from the objective

Table 8.6 Comparison 2 CPU Time of MATLAB vs IMSL

γ	MATLAB CPU sec	IMSL CPU sec
95	43.95	153.33
90	27.41	20.05
85	148.86	17.93
80	20.53	18.56
75	16.84	18.11
70	18.66	18.01
65	17.04	20.43
60	17.45	29.38
55	18.18	26.65
50	16.16	37.21
45	17.05	25.13
40	19.63	20.03
35	17.66	24.18
30	16.26	166.05
25	16.69	249.13
20	18.03	18.15
15	16.45	21.03
10	22.59	24.86
5	22.81	260.88
TOTAL	512.35	1169.1

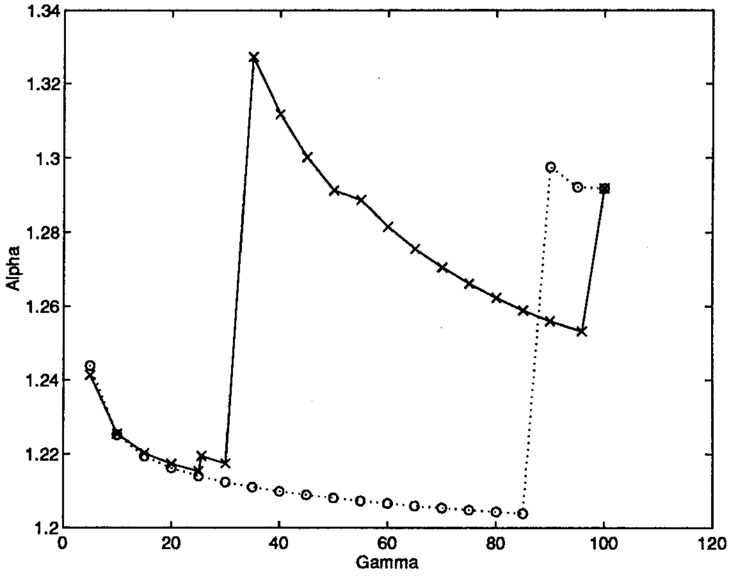

Figure 8.6 Mixed H_2/H_∞ Curve MATLAB-'o' IMSL-'x' (Comparison 2)

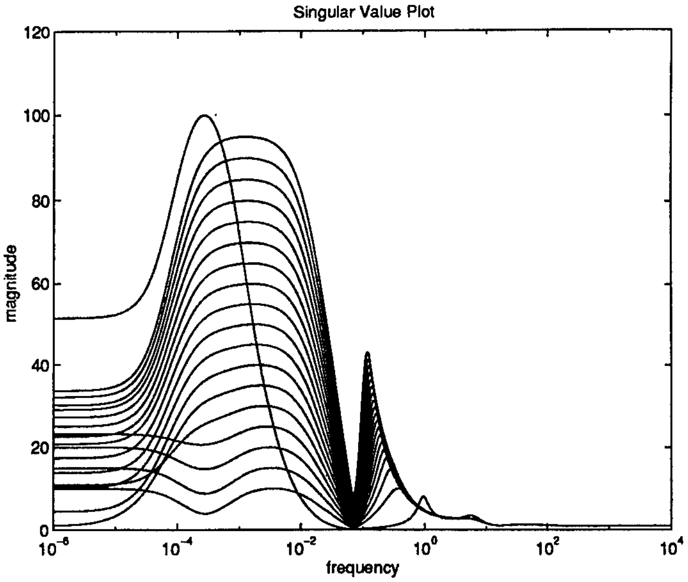

Figure 8.7 Singular Value Plot No Scale IMSL (Comparison 2)

function for IMSL, caused IMSL to find the same local minimum as MATLAB found much earlier. Since the scale factor on the objective was removed, the convergence criteria was set to 10^{-7} to allow IMSL to converge in 20 steps in the line search. A plot of the singular value plot using IMSL with no scale factor on the objective is shown in Figure 8.7. IMSL then took one γ step to achieve the same singular value plot shape as MATLAB. A plot of the mixed H_2/H_∞ curve with the MATLAB curve and the IMSL curve with no scale on the objective is shown in Figure . IMSL and MATLAB compare much better. Finally, without the scale factor for IMSL tables of function evaluations and time comparisons are shown in 8.7 and 8.8.

Notice that even though the scale factor has been taken off the objective for IMSL, the IMSL runs took less time and less function evaluations than the previous IMSL run with the scales. This can be attributed to finding the lower local minimum earlier with the scale than without the scale. The final conclusion from this comparison is that without the scale factor on the objective for IMSL, the mixed curve for MATLAB and IMSL compared very well. The time and number of function evaluations is still more for IMSL than MATLAB.

8.4.3 Comparison 3: Initial Gamma=5, $\Delta\gamma$=-0.25. The final comparison is very close to H_∞ optimal for this problem. Again, $\underline{\gamma}$ for this problem is 1.32. This comparison is

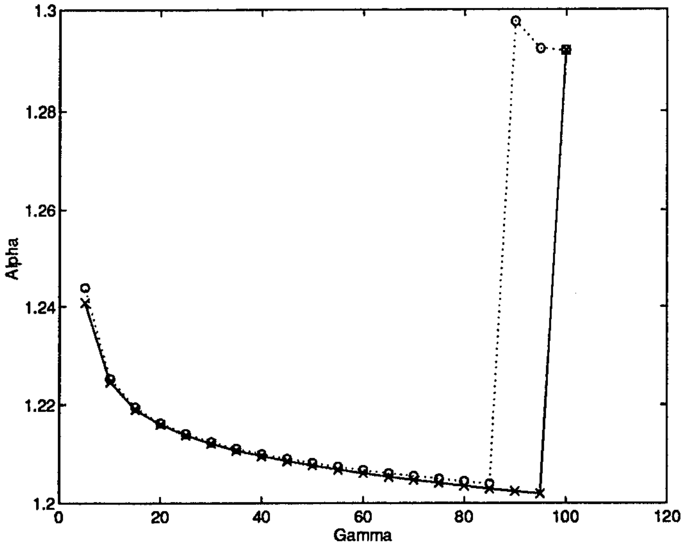

Figure 8.8 Mixed H_2/H_∞ Curve MATLAB-'o' IMSL No Scale-'x' (Comparison 2)

Table 8.7 Comparison 2 Number of Function and Gradient Evaluation

	MATLAB		IMSL	
γ	NUM FUN	NUM GRAD	NUM FUN	NUM GRAD
95	41	20	174	88
90	28	10	27	11
85	178	118	103	28
80	12	8	29	12
75	9	4	27	10
70	10	6	24	12
65	6	6	26	10
60	9	4	27	12
55	8	6	28	11
50	7	4	37	20
45	7	6	27	13
40	11	7	25	12
35	5	4	27	15
30	4	4	37	19
25	4	4	41	20
20	5	5	41	25
15	4	4	28	16
10	8	8	27	16
5	8	8	33	24
TOTAL	364	236	840	374

Table 8.8 Comparison 2 CPU Time of MATLAB vs IMSL

	MATLAB	IMSL
γ	CPU sec	CPU sec
95	43.95	153.43
90	27.41	30.20
85	148.86	86.90
80	20.53	32.38
75	16.84	30.96
70	18.66	29.76
65	17.04	30.16
60	17.45	32.83
55	18.18	31.83
50	16.16	42.68
45	17.05	34.03
40	19.63	32.05
35	17.66	40.76
30	16.26	50.26
25	16.69	53.30
20	18.03	57.65
15	16.45	42.20
10	22.59	42.30
5	22.81	49.10
TOTAL	512.35	909.7

down to γ = 1.5. Due to four peaks being constrained at the start of the example and up to 10 peaks total per iteration being constrained, the IMSL convergence tolerance was relaxed to 10^{-4}. This was the best comparison possible for convergence criteria. Both cases completed all 19 iterations. The singular value plots for MATLAB and IMSL are approximately the same shape except at the frequencies between 10^{-6} and 10^{-2}. The IMSL plot shows a different design being caused by a local minimum at the lowest value of γ. However, the lowest γ singular value curve matches for IMSL and MATLAB. They are shown in Figures 8.9 and 8.10.

The number of function and gradient evaluations is presented in Table 8.9. The IMSL routine required 200 less function evaluations and 100 less gradient evaluations than MAT-

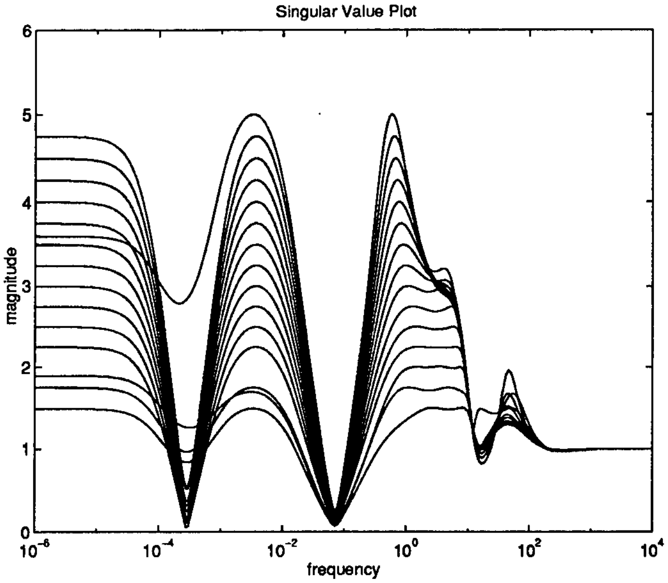

Figure 8.9 Singular Value Plot MATLAB (Comparison 3)

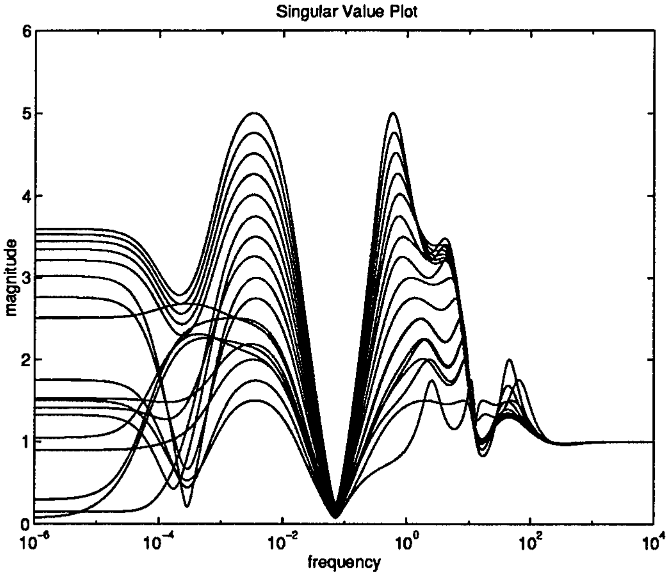

Figure 8.10 Singular Value Plot IMSL (Comparison 3)

LAB. A comparison of CPU time for each step of γ is presented in Table 8.10. The time was reduced by over 1000 CPU seconds by using IMSL.

The mixed H_2/H_∞ curves are shown in Figure 8.11. The local minimum dip in the mixed curve for IMSL is evident at the lowest γ level found. The two-norm and infinity-norm match at the lowest γ value for both the mixed curves of IMSL and MATLAB.

Table 8.9 Comparison 3 Number of Function and Gradient Evaluation Comparison

	MATLAB		IMSL	
γ	NUM FUN	NUM GRAD	NUM FUN	NUM GRAD
4.75	18	18	2	2
4.50	6	6	2	2
4.25	4	4	2	2
4.00	17	17	2	2
3.75	6	6	5	4
3.50	6	6	3	3
3.25	7	7	10	6
3.00	37	32	6	5
2.75	22	15	100	35
2.50	32	26	25	11
2.25	32	27	28	12
2.00	204	37	38	16
1.75	155	64	55	21
1.5	42	21	63	26
TOTAL	588	286	341	147

Table 8.10 Comparison 3 CPU Time of MATLAB vs IMSL

	MATLAB	IMSL
γ	CPU sec	CPU sec
4.75	49.64	19.78
4.50	19.33	13.22
4.25	16.48	13.06
4.00	34.73	13.44
3.75	19.85	17.60
3.50	19.64	15.17
3.25	20.90	23.96
3.00	71.76	20.21
2.75	44.44	133.05
2.50	63.65	50.43
2.25	73.30	56.25
2.00	305.15	81.39
1.75	422.43	102.31
1.5	465.20	127.57
TOTAL	1626.50	687.44

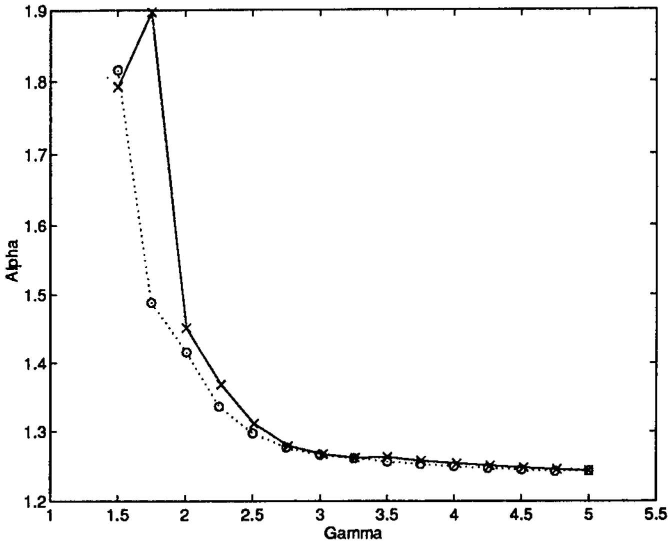

Figure 8.11 Mixed H_2/H_∞ Curve MATLAB-'o' IMSL-'x' (Comparison 3)

8.5 Summary and Conclusions

The number of function and gradient evaluations for each of the three cases is significantly less by approximately (100 to 200 evaluations) for IMSL than MATLAB. The time required to calculate the mixed controller is, in general, less for IMSL than MATLAB. This can be attributed to less function and gradient calculations and the use of FORTRAN code which is, in general, faster than MATLAB. The singular value plots were shown to compare the general trends of the peaks using each method. They were generally the same using either method, but there were examples of different designs being found by either optimizer. The mixed H_2/H_∞ curves were generally smoother for IMSL.

The first comparison showed the IMSL curve to be smooth with no local minimum jumps. However, it was not monotonically increasing. The MATLAB curve showed a monotonically increasing curve with a local minimum dip at the left of the curve. The values of the two-norm were approximately the same in the range of γ that could be compared. The second comparison showed IMSL's mixed curve as a smooth monotonically increasing function. The MATLAB plot showed significant local minimum dips in the curve. The other observation is that the MATLAB curve demonstrated lower values of the two-norm for γ

between (5 to 180). The last comparison demonstrated very similar mixed H_2/H_∞ curves for both MATLAB and IMSL. They were both monotonically increasing with approximately the same two-norm value at different γ levels. The IMSL plot did show a dip in the mixed curve due to a local minimum, but ended at the same design point as MATLAB.

The conclusion from this comparison is that the IMSL/DNCONG routine is a better optimizer than the MATLAB 'CONSTR' function for mixed H_2/H_∞ optimization. It is faster and takes less function and gradient evaluations. It also helps to calculate a smoother mixed H_2/H_∞ curve but does demonstrate local minimum jumps. The second comparison, where the MATLAB curve showed a lower two-norm, means that the MATLAB program found a lower local minimum than the IMSL program. Since this is not a convex problem, this is to be expected.

The final conclusion from this chapter is that the mixed H_2/H_∞ problem should be run using the Shittkowski penalty method for the merit function and his method to update the quadratic sub-problem. These are implemented in IMSL but are not easily accessible to the user to make modifications to tolerances and number of evaluations. Also, calling MATLAB from FORTRAN is cumbersome to code and very difficult to debug. The best solution would be to implement the Shittkowski methods in place of MATLAB's 'CONSTR' function. This would improve the speed and convergence of the mixed H_2/H_∞ problem.

IX. Conclusions and Recommendations

9.1 Thesis Summary and Conclusions

This thesis identified ways to improve the run time and the reliability of the mixed H_2/H_∞ optimization process. The improved code was run for a realistic flight control problem of an F-16. The controller design for the realistic F-16 problem was done at fifth, sixth and seventh order. Comparisons were then made between the mixed H_2/H_∞ controllers at different orders to the LQG/LTR design. The improved code was then used to generate mixed H_2/H_∞ controllers for a MIMO missile problem. Mixed H_2/H_∞ curves were found for both the 3x3 MIMO and the 5x3 MIMO designs. Finally, a comparison of the MATLAB 'CONSTR' optimization function and the FORTRAN IMSL/DNCONG optimization routine was done.

The purpose of this thesis was to improve the run time and reliability of the mixed H_2/H_∞ optimization process and then demonstrate its effectiveness on realistic control problems. The examples discussed show that the run time and reliability of generating a mixed H_2/H_∞ controller has significantly improved. The results from this thesis show that the mixed H_2/H_∞ design is better for the F-16 example than the LQG/LTR example even for a controller that is three orders less than the LQG/LTR. It has also demonstrated that where numeric problems for the MIMO missile example prevented mixed H_2/H_∞ designs, these controllers can now be found. Finally, the comparison between the MATLAB and IMSL optimization routines proved that the IMSL optimization method was better for mixed H_2/H_∞ optimization.

9.2 Recommendations

This study has raised several new questions about the mixed H_2/H_∞ optimization design process. The following are suggested avenues of study which the results of this thesis indicate to be worthwhile.

- *Code Shittkowski's optimization methods in MATLAB.* This would provide the designer with the efficiency of the IMSL optimization routine and the user friendliness of MATLAB. It would also provide the ability to make modifications to the optimization code when necessary, such as the change to only evaluate the maximum H_∞ constraint.

- *Investigate way to reduce the probability of finding a local minimum in the mixed H_2/H_∞ optimization process.* There are methods called simulated annealing and genetic algorithms that could be added to this code to improve the chances of the optimization routine moving away from a local minimum. Implementing these methods could improve the mixed H_2/H_∞ curve by improving the chances that a global minimum could be found.

- *Scale the design variables.* This would be done to possibly increase efficiency. The functions, constraints and their gradients were all scaled for this study. The other possibility to improve performance of the optimization process is to scale the design variables.

Appendix A. SISO Example: Matrices for Underlying H_2 and H_∞ Problems

The matrices for the SISO example's underlying H_2 problem are:

$$A_2 = \begin{bmatrix} -1.4850E-02 & 3.7382E+01 & -3.2200E+01 & -1.7940E+01 & 2.1400E-03 & 0.0000E+00 \\ -8.0000E-05 & -1.4910E+00 & -1.3000E-03 & 9.9600E-01 & -1.8800E-01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -3.6000E-04 & 9.7530E+00 & 2.9000E-04 & -9.6000E-01 & -1.9040E+01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -2.0000E+01 & 0.0000E+00 \\ 1.5000E-03 & 3.5264E+01 & 2.7200E-02 & -3.3400E-01 & -4.3660E+00 & -4.0000E+01 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 8.3589E-01 & 0.0000E+00 \\ -3.3340E-02 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \\ 2.1808E-01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$B_{u_2} = \begin{bmatrix} 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 2.0000E+01 \\ 0.0000E+00 \end{bmatrix}$$

$$C_z = \begin{bmatrix} -1.5000E-03 & -3.5264E+01 & -2.7200E-02 & 3.3400E-01 & 4.3660E+00 & 8.0000E+01 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$D_{zw} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$D_{zu} = \begin{bmatrix} 0.0000E+00 \\ 1.0000E+01 \end{bmatrix}$$

$$C_{y_2} = \begin{bmatrix} -1.5000E-03 & -3.5264E+01 & -2.7200E-02 & 3.3400E-01 & 4.3660E+00 & 8.0000E+01 \end{bmatrix}$$

$$D_{yw} = \begin{bmatrix} 0.0000E+00 & 4.0000E-03 \end{bmatrix}$$

$$D_{yu} = \begin{bmatrix} 0.0000E+00 \end{bmatrix}$$

The matrices for the SISO example's underlying H_∞ problem are:

$$A_\infty = \begin{bmatrix} -1.4850E-02 & 3.7382E+01 & -3.2200E+01 & -1.7940E+01 & 2.1400E-03 & 0.0000E+00 & 0.0000E+00 \\ -8.0000E-05 & -1.4910E+00 & -1.3000E-03 & 9.9600E-01 & -1.8800E-01 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -3.6000E-04 & 9.7530E+00 & 2.9000E-04 & -9.6000E-01 & -1.9040E+01 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -2.0000E+01 & 0.0000E+00 & 0.0000E+00 \\ 1.5000E-03 & 3.5264E+01 & 2.7200E-02 & -3.3400E-01 & -4.3660E+00 & -4.0000E+01 & 0.0000E+00 \\ -1.5000E-03 & -3.5264E+01 & -2.7200E-02 & 3.3400E-01 & 4.3660E+00 & 8.0000E+01 & -1.0000E-04 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 0.0000E+00 \\ 1.0000E+00 \end{bmatrix}$$

$$B_{u_\infty} = \begin{bmatrix} 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 \\ 2.0000\mathrm{E}+01 \\ 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 \end{bmatrix}$$

$$C_e = \begin{bmatrix} -1.5000\mathrm{E}-03 & -3.5264\mathrm{E}+01 & -2.7200\mathrm{E}-02 & 3.3400\mathrm{E}-01 & 4.3660\mathrm{E}+00 & 8.0000\mathrm{E}+01 & 1.0000\mathrm{E}+00 \end{bmatrix}$$

$$D_{ed} = \begin{bmatrix} 1.0000\mathrm{E}+00 \end{bmatrix}$$

$$D_{eu} = \begin{bmatrix} 0.0000\mathrm{E}+00 \end{bmatrix}$$

$$C_{y_\infty} = \begin{bmatrix} -1.5000\mathrm{E}-03 & -3.5264\mathrm{E}+01 & -2.7200\mathrm{E}-02 & 3.3400\mathrm{E}-01 & 4.3660\mathrm{E}+00 & 8.0000\mathrm{E}+01 & 0.0000\mathrm{E}+00 \end{bmatrix}$$

$$D_{yd} = \begin{bmatrix} 1.0000\mathrm{E}+00 \end{bmatrix}$$

Appendix B. 3x3 MIMO Problem: Matrices for Underlying H_2 and H_∞ Problems

The matrices for the 3x3 MIMO example's underlying H_2 problem are:

$$A_2 \quad = \quad \left[\begin{array}{cccc} -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & -4.0000E+02 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & -4.0000E+02 & \ldots \\ -8.7451E-01 & 1.3044E+02 & -4.5412E-01 & \ldots \\ -2.4756E-01 & 6.2620E-01 & -1.2753E+02 & \ldots \\ 4.5574E+06 & 1.2009E+06 & -4.2461E+03 & \ldots \\ -1.0943E+03 & 1.2844E+05 & 5.8025E+01 & \ldots \\ -1.2180E+02 & 5.6387E+02 & 1.2598E+05 & \ldots \end{array} \right.$$

$$\left. \begin{array}{ccccc} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -9.8435E-01 & -9.2340E-02 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ -9.2340E-02 & -9.8435E-01 & 0.0000E+00 & 0.0000E+00 & -1.0000E+00 \\ -2.6737E+02 & 2.6737E+02 & -1.5949E+00 & 0.0000E+00 & 0.0000E+00 \\ -1.9462E+02 & -6.9672E-01 & 0.0000E+00 & -1.5498E+00 & 0.0000E+00 \\ 6.9673E-01 & 1.9462E+02 & 0.0000E+00 & 0.0000E+00 & -1.5498E+00 \end{array} \right]$$

$$B_w \quad = \quad \left[\begin{array}{ccccc} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -1.9687E-03 & -1.8468E-05 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -1.8468E-04 & -1.9687E-04 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -5.3474E-01 & 5.3474E-02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -3.8924E-01 & -1.3934E-04 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 1.3935E-03 & 3.8924E-02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{array} \right]$$

$$B_{u_2} = \begin{bmatrix} 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$C_z = \left[\begin{matrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ -5.4953E+02 & 1.3982E+03 & -2.8476E+05 & \ldots \\ -1.9504E+03 & 2.9126E+05 & -1.0140E+03 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \end{matrix} \right.$$

$$\left. \begin{matrix} 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.0619E+02 & -2.1979E+03 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.1979E+03 & -2.0619E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{matrix} \right]$$

$$D_{zw} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$D_{zu} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 7.0711E+01 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 7.0711E+01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 7.0711E+01 \end{bmatrix}$$

$$C_{y_2} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ -5.4953E+02 & 1.3982E+03 & -2.8476E+05 & \ldots \\ -1.9504E+03 & 2.9126E+05 & -1.0140E+03 & \ldots \end{bmatrix}$$

$$\begin{matrix} 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.0619E+02 & -2.1979E+03 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.1979E+03 & -2.0619E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{matrix} \Bigg]$$

$$D_{yw} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 5.4538E-05 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.2500E-03 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.2500E-03 \end{bmatrix}$$

$$D_{yu} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

The matrices for the 3x3 MIMO example's underlying H_∞ problem are:

$$A_\infty \;=\; \left[\begin{array}{cccccc}
-4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & 0.0000E+00 & -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & \ldots \\
-8.7451E-01 & 1.3044E+02 & -4.5412E-01 & -9.8435E-01 & -9.234E-02 & \ldots \\
-2.4756E-01 & 6.2620E-01 & -1.2753E+02 & -9.2340E-02 & -9.8435E-01 & \ldots \\
4.5574E+06 & 1.2009E+06 & -4.2461E+03 & -2.6737E+02 & 2.6737E+02 & \ldots \\
-1.0943E+03 & 1.2844E+05 & 5.8025E+01 & -1.9462E+02 & -6.9672E-01 & \ldots \\
-1.2180E+02 & 5.6387E+02 & 1.2598E+05 & 6.0673E-01 & 1.9462E+02 & \ldots \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
-5.4736E+02 & 1.3982E+03 & -2.8476E+05 & -2.0619E+02 & -2.1979E+03 & \ldots \\
-1.9502E+03 & 2.9126E+05 & -1.0140E+03 & -2.1979E+03 & -2.0619E+02 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{cccccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & -1.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-1.5949E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & -1.5498E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & -1.5498E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
1.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.0000E-04 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.0000E-04 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.0000E-04
\end{array}\right]$$

$$B_d \;=\; \left[\begin{array}{ccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 1.0000E+00
\end{array}\right]$$

$$B_{u_\infty} = \begin{bmatrix} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \end{bmatrix}$$

$$C_e = \left[\begin{matrix} 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \dots \\ -5.4953\mathrm{E}+02 & 1.3982\mathrm{E}+03 & -2.8476\mathrm{E}+05 & -2.0619\mathrm{E}+02 & -2.1979\mathrm{E}+03 & \dots \\ -1.9502\mathrm{E}+03 & 2.9126\mathrm{E}+05 & -1.0140\mathrm{E}+03 & -2.1979\mathrm{E}+03 & -2.0619\mathrm{E}+02 & \dots \end{matrix} \right.$$

$$\left. \begin{matrix} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 2.0000\mathrm{E}+01 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 2.0000\mathrm{E}+01 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 2.0000\mathrm{E}+01 \end{matrix} \right]$$

$$D_{ed} = \begin{bmatrix} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 \end{bmatrix}$$

$$D_{eu} = \begin{bmatrix} 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \end{bmatrix}$$

$$C_{y_\infty} = \left[\begin{matrix} 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \dots \\ -5.4736\mathrm{E}+02 & 1.3982\mathrm{E}+03 & -2.8476\mathrm{E}+05 & -2.0619\mathrm{E}+02 & -2.1979\mathrm{E}+03 & \dots \\ -1.9502\mathrm{E}+03 & 2.9126\mathrm{E}+05 & -1.0140\mathrm{E}+03 & -2.1979\mathrm{E}+03 & -2.0619\mathrm{E}+02 & \dots \end{matrix} \right.$$

$$\left. \begin{matrix} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \end{matrix} \right]$$

$$D_{yd} = \begin{bmatrix} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 \end{bmatrix}$$

Appendix C. 5x3 MIMO Problem: Matrices for Underlying H_2 and H_∞ Problems

The matrices for the 5x3 MIMO example's underlying H_2 problem are:

$$A_2 = \left[\begin{array}{ccccc}
-4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & 0.0000E+00 & -4.0000E+02 & 0.0000E+00 & \ldots \\
-8.7349E-01 & 1.3044E+02 & -4.5412E-01 & -9.8435E-01 & \ldots \\
-2.4611E-01 & 6.2620E-01 & -1.2753E+02 & -9.2340E-02 & \ldots \\
3.3304E+06 & 1.2009E+06 & -4.2461E+03 & -2.6737E+02 & \ldots \\
-1.0950E+03 & 1.2844E+05 & 5.8025E+01 & -1.9462E+02 & \ldots \\
-1.2044E+02 & 5.6387E+02 & 1.2598E+05 & 6.9673E-01 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{cccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-9.2340E-02 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\
-9.8435E-01 & 0.0000E+00 & 0.0000E+00 & -1.0000E+00 \\
2.6737E+02 & -1.5949E+00 & 0.0000E+00 & 0.0000E+00 \\
-6.9672E-01 & 0.0000E+00 & -1.5498E+00 & 0.0000E+00 \\
1.9462E+02 & 0.0000E+00 & 0.0000E+00 & -1.5498E+00
\end{array}\right]$$

$$B_w = \left[\begin{array}{ccccccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-1.9687E-03 & -1.8468E-05 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-1.8468E-04 & -1.9687E-04 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-5.3474E-01 & 5.3474E-02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-3.8924E-01 & -1.3934E-04 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
1.3935E-03 & 3.8924E-02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00
\end{array}\right]$$

$$B_{u_2} \quad = \quad \begin{bmatrix} 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$C_z \quad = \quad \left[\begin{array}{cccc} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ -5.4953E+02 & 1.3982E+03 & -2.8476E+05 & \ldots \\ -1.9504E+03 & 2.9126E+05 & -1.0140E+03 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \end{array} \right.$$

$$\left. \begin{array}{ccccc} 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ -2.0619E+02 & -2.1979E+03 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.1979E+03 & -2.0619E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{array} \right]$$

$$D_{zw} \quad = \quad \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$D_{zu} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 7.0711E+01 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 7.0711E+01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 7.0711E+01 \end{bmatrix}$$

$$C_{y_2} = \left[\begin{array}{rrrc} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\ -5.4953E+02 & 1.3982E+03 & -2.8476E+05 & \ldots \\ -1.9504E+03 & 2.9126E+05 & -1.0140E+03 & \ldots \end{array} \right.$$

$$\left. \begin{array}{rrrrr} 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ -2.0619E+02 & -2.1979E+03 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ -2.1979E+03 & -2.0619E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{array} \right]$$

$$D_{yw} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 5.4538E-05 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 5.4538E-05 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 5.4538E-05 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 4.0250E-02 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 4.0250E-02 \end{bmatrix}$$

$$D_{yu} = \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

The matrices for the 3x5 MIMO example's underlying H_∞ problem are:

$$A_\infty = \left[\begin{array}{ccccccc}
-4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & 0.0000E+00 & -4.0000E+02 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
-8.7349E-01 & 1.3044E+02 & -4.5412E-01 & -9.8435E-01 & -9.2340E-02 & 0.0000E+00 & \ldots \\
-2.4611E-01 & 6.2620E-01 & -1.2753E+02 & -9.2340E-02 & -9.8435E-01 & 0.0000E+00 & \ldots \\
3.3304E+06 & 1.2009E+06 & -4.2461E+03 & -2.6737E+02 & 2.6737E+02 & -1.5949E+00 & \ldots \\
-1.0950E+03 & 1.2844E+05 & 5.8025E+01 & -1.9462E+02 & -6.9672E-01 & 0.0000E+00 & \ldots \\
-1.2044E+02 & 5.6387E+02 & 1.2598E+05 & 6.9673E-01 & 1.9462E+02 & 0.0000E+00 & \ldots \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 2.0000E+01 & \ldots \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & \ldots \\
-1.0991E+04 & 2.7964E+04 & -5.6952E+06 & -4.1238E+03 & -4.3958E+04 & 0.0000E+00 & \ldots \\
-3.9008E+04 & 5.8252E+06 & -2.0280E+04 & -4.3958E+04 & -4.1238E+03 & 0.0000E+00 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{ccccccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
1.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & -1.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
-1.5498E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & -1.5498E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & -1.0000E-04 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.0000E-04 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.0000E-04
\end{array}\right]$$

$$B_d \quad = \quad \begin{bmatrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 2.0000E+01 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 2.0000E+01 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 2.0000E+01 \end{bmatrix}$$

$$B_{u_\infty} \quad = \quad \begin{bmatrix} 1.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{bmatrix}$$

$$C_e = \left[\begin{array}{ccccccc}
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & \ldots \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \ldots \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \ldots \\
-5.4953\mathrm{E}+02 & 1.3982\mathrm{E}+03 & -2.8476\mathrm{E}+05 & -2.0619\mathrm{E}+02 & -2.1979\mathrm{E}+03 & 0.0000\mathrm{E}+00 & \ldots \\
-1.9504\mathrm{E}+03 & 2.9126\mathrm{E}+05 & -1.0140\mathrm{E}+03 & -2.1979\mathrm{E}+03 & -2.0619\mathrm{E}+02 & 0.0000\mathrm{E}+00 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{ccccccc}
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00
\end{array}\right]$$

$$D_{ed} = \begin{bmatrix}
1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00
\end{bmatrix}$$

$$D_{eu} = \begin{bmatrix}
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\
0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00
\end{bmatrix}$$

$$C_{y_\infty} = \left[\begin{array}{ccccccc} 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & \ldots \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \ldots \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & \ldots \\ -5.4953\mathrm{E}+02 & 1.3982\mathrm{E}+03 & -2.8476\mathrm{E}+05 & -2.0619\mathrm{E}+02 & -2.1979\mathrm{E}+03 & 0.0000\mathrm{E}+00 & \ldots \\ -1.9504\mathrm{E}+03 & 2.9126\mathrm{E}+05 & -1.0140\mathrm{E}+03 & -2.1979\mathrm{E}+03 & -2.0619\mathrm{E}+02 & 0.0000\mathrm{E}+00 & \ldots \end{array} \right.$$

$$\left. \begin{array}{ccccccc} 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \end{array} \right]$$

$$D_{yd} = \left[\begin{array}{ccccc} 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 \\ 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 0.0000\mathrm{E}+00 & 1.0000\mathrm{E}+00 \end{array} \right]$$

Bibliography

ASC83. A.N. Andry, Jr., E.Y. Shapiro, and J.C. Chung. Eigenstructure assignment for linear systems. *IEEE Transactions on Automatic Control*, AES-19(5):711–729, September 1983.

Bai92. James C. Baird. Strategies for optimal control design of normal acceleration command following on the f-16. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, December 1992.

BH89. D. S. Bernstein and W. H. Haddad. Lqg control with an H_∞ performance bound: A riccati equation approach. *IEEE Transactions on Automatic Control*, AC-34(3):293–305, March 1989.

Bro91. James Milton Brown II. Design of an air-to-air missile autopilot using linear quadratic gaussian/loop transfer recovery. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, December 1991.

Can94. R. Canfield. Analytic gradient of two-norm. *Correspondence*, August 1994.

DGKF89. John C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, AC-34(8):831–847, August 1989.

DS79. John C. Doyle and G. Stein. Robustness with observers. *IEEE Transactions on Automatic Control*, AC-24:607–611, August 1979.

GL93. D. P. Geisy and K. B. Lim. H_∞ norm sensitivity formula with control system design applications. *Journal of Guidance, Control and Dynamics*, 16(6):1138–1145, 1993.

Gra93. Andrew Grace. *MATLAB Optimization Toolbox.* The MathWorks, Inc, Massachusetts, 1993.

IMS89. IMSL. *IMSL Math/Library Users Manual.* IMSL, Texas, 1989.

KR91. P. P. Khargonekar and M. A. Rotea. Mixed H_2/H_∞ control: A convex optimization approach. *IEEE Transactions on Automatic Control*, AC-36(7):824–837, July 1991.

Luk93. James P. Luke. Flight controller design using mixed H_2/H_∞ optimization with a singular H_∞ constraint. Master's thesis, Air Force Institute of Technology, Dayton, OH 45433, December 1993.

Rid91. D. B. Ridgely. *A Nonconservative Solution to the General Mixed H_2/H_∞ Optimization Problem.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1991.

Sch85. K. Schittkowski. Nlpqp: A fortran subroutine solving constrained nonlinear programming problems. *Anals of Operation Research*, 6:485–500, 1985.

Van84. Garret Vanderplaats. *Numerical Optimization Techniques for Engineering Design.* McGraw-Hill, Inc., New York, 1984.

Wal94. David E. Walker. *H_2 Optimal Control with H_∞, μ, and L_1 Constraints.* PhD thesis, Air Force Institute of Technology, Dayton, OH, 1994.

WR92. S. R. Wells and D. B. Ridgely. Using increased order controllers in mixed H_2/H_∞ optimization. In *Proceedings of American Control Conference*, pages 1358–1362, Chicago, IL, June 1992.

YBC90. H.H. Yeh, S.S. Banda, and B.C. Chang. Necessary and sufficient conditions for mixed H_2 and H_∞ optimal control. In *Proceedings of the 29th Conference on Decision and Control*, pages 1013–1017, Honolulu, HI, December 1990.

Zam89. G. Zames. On the input-output stability of time-varying nonlinear feedback systems, part i: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, AC-11:228–238, 1989.

ZDB89. K. Zhou, J. Doyle, and B. Bodenheimer. Optimal control with mixed H_2 and H_∞ performance objectives. In *Proceedings of American Control Conference*, pages 2065–2070, Pittsburgh, PA, June 1989.

ZDGB90. K. Zhou, J. Doyle, K. Glover, and B. Bodenheimer. Mixed H_2 and H_∞ control. In *Proceedings of the American Control Conference*, pages 2502–2507, San Diego, CA, May 1990.

Vita

Capt Linda Smith was born on 16 December 1965 in Concordia, Kansas. She graduated from Saint Mary's High School, Independence, Missouri in 1984 and attended the University of Missouri-Rolla on an AFROTC scholarship. Capt Smith was commissioned in December of 1988 and served her first assignment as a system test engineer. She worked as a test engineer for both the Advanced Medium Range Air-to-Air Joint System Program Office and then a one year High Gear demonstration program for the Joint Direct Attack Munitions System Program Office, Eglin AFB, FL. Capt Smith entered the Air Force Institute of Technology's Graduate School of Engineering in May 1993, and upon completion of her studies, will be assigned to the Flight Dynamics Lab, Wright-Patterson AFB, OH.

Permanent address: 9140 Brandy Court
Spring Valley, Ohio 45370...

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1994

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
IMPROVED NUMERICAL SOLUTION OF MIXED H_2/H_∞ OPTIMIZATION WITH APPLICATIONS

5. FUNDING NUMBERS

6. AUTHOR(S)
Linda D. Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAE/ENY/94D-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr Marc Jacobs
AFOSR/NM
110 Duncan Ave, Suite B115
Bolling AFB DC 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release;
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This study improved the numerical optimization technique used to compute mixed H_2/H_∞ controllers. The run time to generate the full curve of an H_2/H_∞ design was reduced by a factor of almost 200 for a specific example. This was accomplished by improving gradient calculations, upgrading the H_∞ norm calculation to capture multiple peaks in the singular value plot and using routines that minimized processing time. The optimization routine used was the MATLAB 'CONSTR' Sequential Quadratic Programming (SQP) routine. This SQP routine was compared to a FORTRAN SQP numerical optimizer, IMSL/DNCONG. The method was applied to an F-16 SISO example and a missile MIMO example. The mixed H_2/H_∞ controller for both these examples are compared to previous controller designs created using the LQG/LTR technique.

14. SUBJECT TERMS
Control Theory, Mathematical Programming, H2 Optimization, H-inf Optimization, Numerical Optimization

15. NUMBER OF PAGES
169

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.